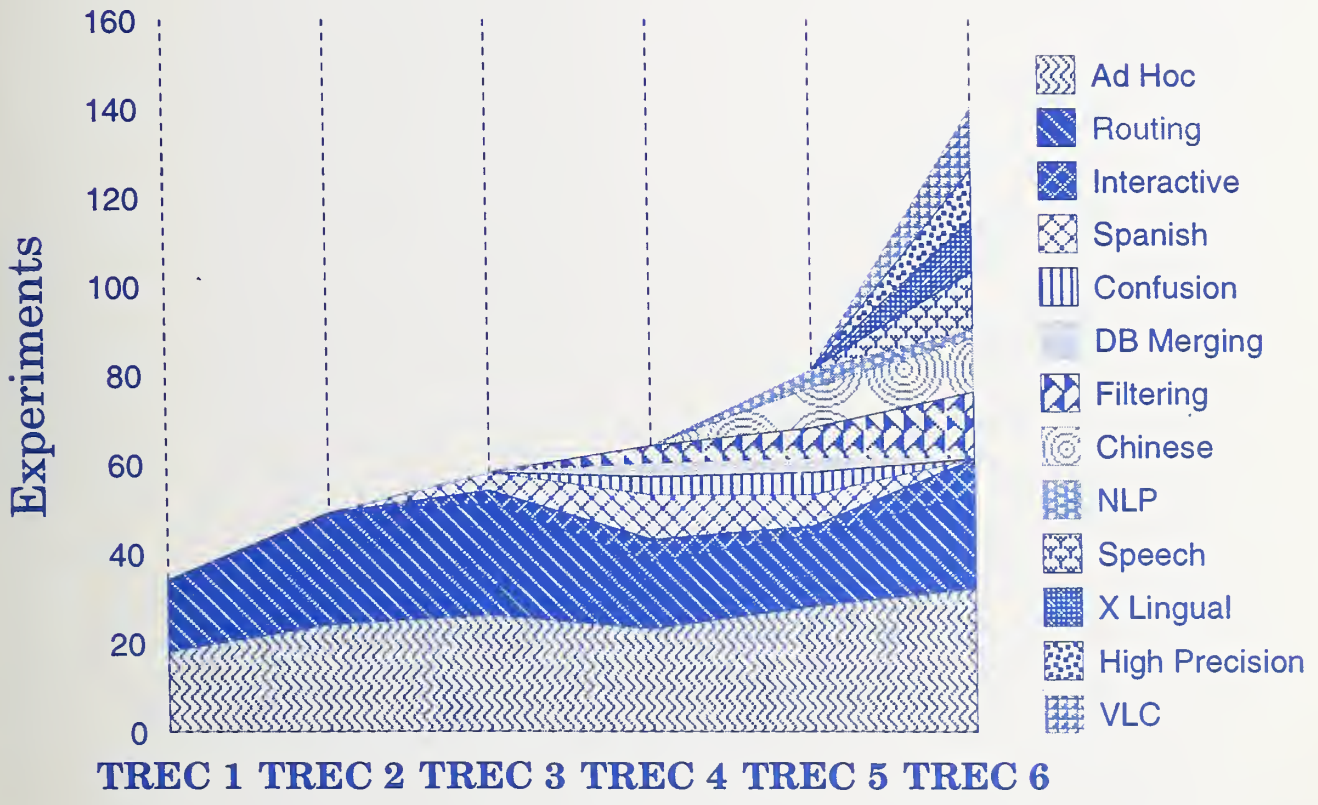
NIST Special Publication 500-240

Information Technology:

The Sixth Text REtrieval Conference (TREC-6)

E. M. Voorhees and
D. K. Harman, Editors



U.S. Department of Commerce
Technology Administration
National Institute of
Standards and Technology

NIST

The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering, and develops measurement techniques, test methods, standards, and related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information contact the Publications and Program Inquiries Desk, 301-975-3058.

Office of the Director
- National Quality Program
- International and Academic Affairs

Technology Services
- Standards Services
- Technology Partnerships
- Measurement Services
- Technology Innovation
- Information Services

Advanced Technology Program
- Economic Assessment
- Information Technology and Applications
- Chemical and Biomedical Technology
- Materials and Manufacturing Technology
- Electronics and Photonics Technology

Manufacturing Extension Partnership Program
- Regional Programs
- National Programs
- Program Development

Electronics and Electrical Engineering Laboratory
- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Electromagnetic Fields[1]
- Electromagnetic Technology[1]
- Optoelectronics[1]

Chemical Science and Technology Laboratory
- Biotechnology
- Physical and Chemical Properties[2]
- Analytical Chemistry
- Process Measurements
- Surface and Microanalysis Science

Physics Laboratory
- Electron and Optical Physics
- Atomic Physics
- Optical Technology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Materials Science and Engineering Laboratory
- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- NIST Center for Neutron Research

Manufacturing Engineering Laboratory
- Precision Engineering
- Automated Production Technology
- Intelligent Systems
- Fabrication Technology
- Manufacturing Systems Integration

Building and Fire Research Laboratory
- Structures
- Building Materials
- Building Environment
- Fire Safety Engineering
- Fire Science

Information Technology Laboratory
- Mathematical and Computational Sciences[2]
- Advanced Network Technologies
- Computer Security
- Information Access and User Interfaces
- High Performance Systems and Services
- Distributed Computing and Information Services
- Software Diagnostics and Conformance Testing

[1] At Boulder, CO 80303.
[2] Some elements at Boulder, CO.

NIST Special Publication 500-240

Information Technology:

The Sixth Text REtrieval Conference (TREC-6)

E. M. Voorhees and
D. K. Harman, Editors

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001

August 1998

U.S. Department of Commerce
William M. Daley, Secretary

Technology Administration
Gary R. Bachula, Acting Under Secretary for Technology

National Institute of Standards and Technology
Raymond G. Kammer, Director

Reports on Information Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) stimulates U.S. economic growth and industrial competitiveness through technical leadership and collaborative research in critical infrastructure technology, including tests, test methods, reference data, and forward-looking standards, to advance the development and productive use of information technology. To overcome barriers to usability, scalability, interoperability, and security in information systems and networks, ITL programs focus on a broad range of networking, security, and advanced information technologies, as well as the mathematical, statistical and computational sciences. This Special Publication 500 series reports on ITL's research in tests and test methods for information technology, and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology
Special Publication 500-240
Natl. Inst. Stand. Technol.
Spec. Publ. 500-240
1038 pages (Aug. 1998)
CODEN: NSPUE2

U.S. Government Printing Office
Washington: 1998

For sale by the Superintendent of Documents
U.S. Government Printing Office, Washington, DC 20402-9325

Foreword

This report constitutes the proceedings of the sixth Text REtrieval Conference (TREC-6) held in Gaithersburg, Maryland, November 19–21, 1997. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA), and was attended by 150 people. Fifty-one groups including participants from 12 different countries and 21 companies were represented. The conference was the sixth in an on-going series of workshops to evaluate new technologies in text retrieval.

The workshop included plenary sessions, discussion groups, a poster session, and demonstrations. Because the participants in the workshop drew on their personal experiences, they sometimes cited specific vendors and commercial products. The inclusion or omission of a particular company or product implies neither endorsement nor criticism by NIST.

The sponsorship of the Information Technology Office of the Defense Advanced Research Projects Agency is gratefully acknowledged, as is the tremendous work of the program committee.

Ellen Voorhees,
Donna Harman
June 24, 1998

TREC-6 Program Committee

Donna Harman, NIST, chair
Nick Belkin, Rutgers University
Chris Buckley, SaBIR Research, Inc.
Jamie Callan, University of Massachusetts at Amherst
Susan Dumais, Microsoft
Darryl Howard, U.S. Department of Defense
David Hull, Xerox Research Centre Europe
David Lewis, AT&T Research
John Prange, U.S. Department of Defense
Steve Robertson, City University, UK
Peter Schäuble, Swiss Federal Institute of Technology (ETH)
Alan Smeaton, Dublin City University, Ireland
Karen Sparck Jones, University of Cambridge, UK
Richard Tong, Tarragon Consulting Corporation
Howard Turtle, West Group
Ellen Voorhees, NIST
Ross Wilkinson, Royal Melbourne Institute of Technology

TABLE OF CONTENTS

APPENDICES

ALPHABETICAL INDEX OF TREC-6 PAPERS BY ORGANIZATION

INDEX OF TREC-6 PAPERS BY TASK/TRACK

ADHOC TASK

ROUTING TASK

CHINESE TRACK

CROSS-LINGUAL TRACK

FILTERING TRACK

INTERACTIVE TRACK

NLP TRACK

SDR TRACK

VLC TRACK

Abstract

This report constitutes the proceedings of the sixth Text REtrieval Conference (TREC-6) held in Gaithersburg, Maryland, November 19–21, 1997. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA), and was attended by 150 people. Fifty-one groups including participants from 12 different countries and 21 companies were represented.

The goal of the conference was to bring research groups together to discuss their work on a large test collection. The diversity of the participants meant that a wide variety of retrieval techniques were represented, including machine learning methods for query expansion and term weighting, sophisticated natural language processing techniques, and advanced pattern matching. Results were scored using a common evaluation package, so groups were able to compare the effectiveness of different techniques, and to discuss how differences between systems affected performance. In addition to the main evaluation, eight additional evaluations, called "tracks," allowed participants to focus on particular common subproblems.

The conference included paper sessions and discussion groups. This proceedings includes papers from most of the participants (some groups did not submit papers), track reports that define the problem addressed by the track plus summarize the main track results, and tables of individual group results. The TREC-6 proceedings web site also contains system descriptions that detail the timing and storage requirements of the different runs.

Overview of the Sixth Text REtrieval Conference (TREC-6)

Ellen M. Voorhees, Donna Harman
National Institute of Standards and Technology
Gaithersburg, MD 20899

1 Introduction

The sixth Text REtrieval Conference (TREC-6) was held at the National Institute of Standards and Technology (NIST) on November 19–21, 1997. The conference was co-sponsored by NIST and the Information Technology Office of the Defense Advanced Research Projects Agency (DARPA) as part of the TIPSTER Text Program.

TREC-6 is the latest in a series of workshops designed to foster research in text retrieval. For analyses of the results of previous workshops, see Sparck Jones [6], Tague-Sutcliffe and Blustein [8], and Harman [2]. In addition, the overview paper in each of the previous TREC proceedings summarizes the results of that TREC.

The TREC workshop series has the following goals:

- to encourage research in text retrieval based on large test collections;
- to increase communication among industry, academia, and government by creating an open forum for the exchange of research ideas;
- to speed the transfer of technology from research labs into commercial products by demonstrating substantial improvements in retrieval methodologies on real-world problems; and
- to increase the availability of appropriate evaluation techniques for use by industry and academia, including development of new evaluation techniques more applicable to current systems.

Table 1 lists the groups that participated in TREC-6. Fifty-one groups including participants from 12 different countries and 21 companies were represented. The diversity of the participating groups has ensured that TREC represents many different approaches to text retrieval. The emphasis on individual experiments evaluated within a common setting has proven to be a major strength of TREC.

This paper serves as an introduction to the research described in detail in the remainder of the volume. The next section defines the common retrieval tasks performed in TREC-6. Sections 3 and 4 provide details regarding the test collections and the evaluation methodology used in TREC. Section 5 provides an overview of the retrieval results. The final section summarizes the main themes learned from the experiments.

2 The Tasks

Each of the TREC conferences has centered around two main tasks, the routing task and the ad hoc task. In addition, starting in TREC-4 a set of "tracks," tasks that focus on particular subproblems of text retrieval, was introduced. TREC-6 continued four tracks from previous years and introduced four new tracks. This section describes the goals of the two main tasks in detail, and outlines the goals of each of the tracks. Readers are urged to consult the appropriate track report found later in these proceedings for details about individual tracks.

2.1 The routing task

The routing task in the TREC workshops investigates the performance of systems that use standing queries to search new streams of documents. These searches are similar to those required by news clipping services and library profiling systems. A true routing environment is simulated in TREC by using topics that have known relevant documents and testing on a completely new document set.

The training for the routing task is shown in the left-hand column of Figure 1. Participants are given a set of topics and a document set that includes known relevant documents for those topics. The topics consist of natural language text describing a user's information need (see sec. 3.2 for details). The topics are used to create a set of queries (the actual input to the retrieval system) that are then used against

Table 1: Organizations participating in TREC-6

Apple Computer	MIT/IBM Almaden Research Center
AT&T Labs Research	NEC Corporation
Australian National University	New Mexico State U. (2 groups)
CEA (France)	NSA (Speech Research Branch)
Carnegie Mellon University	Open Text Corporation
Center for Information Research, Russia	Oregon Health Sciences U.
City University, London	Queens College, CUNY
CLARITECH Corporation	Rutgers University (2 groups)
Cornell U./SaBIR Research, Inc	Siemens AG
CSIRO (Australia)	SRI International
Daimler Benz Research Center Ulm	Swiss Federal Inst. of Tech.(ETH)
Dublin City University	TwentyOne (TNO/U-Tente/DFKI/Xerox/U-Tuebingen)
Duke U./U. of Colorado/Bellcore	U. of California, Berkeley
FS Consulting, Inc.	U. of California, San Diego
GE Corp./Rutgers U.	U. of Glasgow
George Mason U./NCR Corp.	U. of Maryland, College Park
Harris Corp.	U. of Massachusetts, Amherst
IBM T.J. Watson Research (2 groups)	U. of Montreal
ITI (Singapore)	U. of North Carolina (2 groups)
MSI/IRIT/U. Toulouse (France)	U. of Sheffield/U. of Cambridge
ISS (Singapore)	U. of Waterloo
APL, Johns Hopkins University	Verity, Inc.
Lexis-Nexis	Xerox Research Centre Europe
MDS at RMIT, Australia	

the training documents. This is represented by Q1 in the diagram. Many Q1 query sets might be built to help adjust the retrieval system to the task, to create better weighting algorithms, and to otherwise prepare the system for testing. The result of the training is query set Q2, routing queries derived from the 47 routing topics and run against the test documents.

The testing phase of the routing task is shown in the middle column of Figure 1. The output of running Q2 against the test documents is the official test result for the routing task. Due to the difficulty of obtaining appropriate data, the test and training documents were not well-matched in both TREC-4 and TREC-5. Since we wanted a good match for TREC-6, we used (mostly) the same routing topics as were used in TREC-5 for TREC-6, and obtained additional Foreign Broadcast Information Service (FBIS) documents as the test set. In particular, we included those TREC-5 routing topics that had at least six relevant documents in the TREC-5 FBIS data as TREC-6 routing topics. Additional relevance assessments were made on the TREC-5 FBIS corpus for several other topics deemed likely to have relevant documents in FBIS, and for four new topics specifically created for the track (topics 10001–10004). For these topics, the top 100 FBIS documents as retrieved by NIST's PRISE search engine were judged and those with at least six relevant were also included in the set of routing topics. The final set of routing topics contained 47 topics.

2.2 The ad hoc task

The ad hoc task investigates the performance of systems that search a static set of documents using new topics. This task is similar to how a researcher might use a library—the collection is known but the questions likely to be asked are not known. The right-hand column of Figure 1 depicts how the ad hoc task is accomplished in TREC. Participants are given approximately 2 gigabytes worth of documents. They are also given 50 new topics. The set of relevant documents for these topics in the document set is not known at the time the participants receive the topics. Participants produce a new query set, Q3, from the ad hoc topics and run those queries against the ad hoc documents. The output from this run is the official test result for the ad hoc task. Topics 301–350

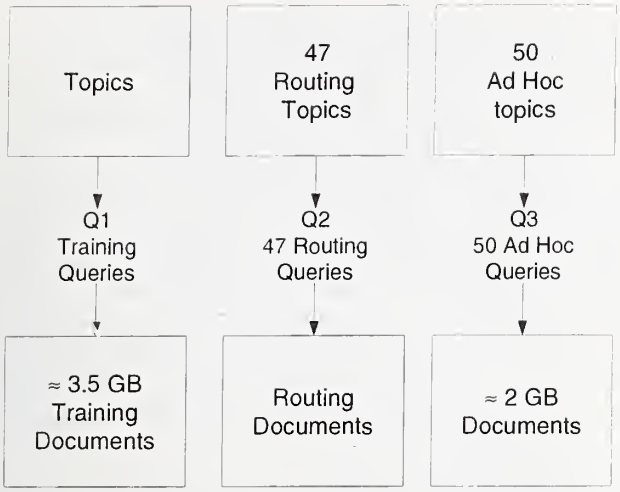


Figure 1: TREC main tasks.

were created for the TREC-6 ad hoc task. The set of documents used in the task were those contained on TREC Disks 4 and 5. See Section 3.1 for details about this document set.

2.3 Task guidelines

In addition to the task definitions, TREC participants are given a set of guidelines outlining acceptable methods of indexing, knowledge base construction, and generating queries from the supplied topics. In general, the guidelines are constructed to reflect an actual operational environment and to allow fair comparisons among the diverse query construction approaches. The allowable query construction methods in TREC-6 are divided into *automatic* methods, in which queries are derived completely automatically from the topic statements, and *manual* methods, which includes queries generated by all other methods. As in TREC-5, the definition of manual query construction methods in TREC-6 permitted users to look at individual documents retrieved by the ad hoc queries and then reformulate the queries based on the documents retrieved.

There are two levels of participation in TREC: category A, participation using the full dataset, or category B, participation using a reduced dataset (1/4 of the full document set). Groups could choose to do the routing task, the ad hoc task, or both, and were asked to submit the top 1000 documents retrieved for each topic for evaluation. Groups that performed the routing task were allowed to submit up to two official test results for judging. When two sets of results were sent, they could be made using different methods of creating queries, or different methods of searching with the same queries. Groups that performed the ad hoc task could submit up to three runs, though if any automatic results were submitted, at least one of the runs was required to use "short" topics (see sec. 3.2).

2.4 The tracks

One of the goals of TREC is to provide a common task evaluation that allows cross-system comparisons. This has proven to be a key strength in TREC. The second major strength is the loose definition of the two main tasks, allowing a wide range of experiments. The addition of secondary tasks (tracks) in TREC-4 combined these strengths by creating a common evaluation for tasks that are either related to the main tasks, or are a more focussed implementation of those tasks. TREC participants were free to turn in results for any, or all, or none, of the tracks. Each track had a set of guidelines developed under the direction of the track coordinator. The set of tracks and their primary goals are listed below. See the track reports elsewhere in this proceedings for a more complete description of each track.

Four tracks continued from previous years and had similar goals as in those years.

Chinese: In the Chinese track, participants performed an ad hoc search in which both the topics and the documents were in Chinese. Twenty-six new topics (CH29–CH54) were created for the track, and the document set was the same as for the TREC-5 track (articles selected from the *Peoples Daily* newspaper and the Xinhua newswire).

Filtering: The filtering task is a routing task in which the system must decide whether or not to retrieve each individual document. Instead of producing a list of documents ranked according to the presumed similarity to a query, filtering systems retrieve an unordered set of documents for each query. The quality of the retrieved set is computed as a function of the benefit of a retrieved relevant document and the cost of a retrieved irrelevant document. The TREC-6 version of the track differed from its predecessors in several ways. New utility functions were introduced to assess the quality of the search. More significantly, filtering track participants could train their systems using only FBIS data (as opposed to all available relevance assessments) and processed the test data in timestamp order.

Interactive: The high-level goal of the interactive track is the investigation of searching as an interactive task by examining the process as well as the outcome. The TREC-6 track used six slightly modified ad hoc topics and the *Financial Times* 1991–1994 collection. The experiment was designed to isolate the effect of topic and searcher from that of the search system and used a common control system to remove other site-specific effects. The searcher task involved six searches (three on control, three on an experimental system) to find and save documents which taken together contained as many answers as possible to the question stated or implied by the topic. System comparisons were based on recall and precision defined in terms of the set of all possible answers as determined by NIST assessors. Participants also reported extensive data on each searcher's interactions with both the control and experimental system.

NLP: The NLP track was initiated to explore whether the natural language processing (NLP) techniques available today are mature enough to have an impact on IR, and specifically whether they can offer an advantage over purely quantitative retrieval methods. The track used the 50 ad hoc topics and the *Financial Times* document set.

The remaining four tracks were introduced in TREC-6.

Cross Language (CLIR): The CLIR task is an ad hoc task in which the focus is on searching for documents in one language using topics in a different language. Three document sets were used in the track: a set of French documents from the Swiss news agency *Schweizerische Depeschen Agentur* (SDA); a set of German documents from SDA plus a set of articles from the newspaper *New Zurich Newspaper* (NZZ); and a set of English documents from the AP newswire. All of the document sets contain news stories from approximately the same time period, but are not aligned or specially coordinated with one another. A set of 25 topics were created by NIST assessors for the track. The authors of the topics created English, French, and German versions of the topics (these were translations of one another). In addition, participants contributed Spanish and Dutch translations of the topics. Participants searched for documents in one target language using topics written in a different language. In addition, participants were asked to perform a monolingual run in the target language to act as a baseline.

High Precision (HP): The goal of the high precision track was to test the effectiveness, efficiency, and user interface of participating systems. Participants used the same 50 topics and document set as the ad hoc task. For each topic, a user was given the query and asked to find 10 documents that answer the topic within 5 minutes (wall clock time). Users could not collaborate on a single topic, nor could the system (or user) have previous knowledge of the topic. Otherwise, the user was free to use any available resources as long as the 5 minute time limit was observed.

Spoken Document Retrieval (SDR): The TREC-6 SDR track was the first running of a track intended to foster research on retrieval methodologies for spoken documents (i.e., recordings of speech). The track is a successor to the "confusion tracks" of earlier TREC conferences, which investigated methods for retrieving document surrogates whose true content has been confused or corrupted in some way. In the SDR track, the document surrogates are produced by speech recognition systems. Participants performed known-item searches using three versions of the documents. The documents were transcripts of radio broadcast news shows: a "truth" transcript that was hand-produced, a transcript produced by a baseline speech recognition system, and a transcript produced by the participant's own speech recognition system.

Very Large Corpus (VLC): The VLC track explored the effectiveness and efficiency of retrieval in collections approximately 10 times the size of a normal TREC collection. The track's corpus consisted of 7.5 million texts for a total of 20.14 GB of data. The TREC-6 ad hoc topics were used. Participants were evaluated on precision of the top 20 retrieved; query response time; data structure building time; and a cost measure of queries/minute/dollar (number of queries processed per minute per hardware dollar).

3 The Test Collections

Like most traditional retrieval collections, there are three distinct parts to the collections used in TREC: the documents, the topics or questions, and the relevance judgments or "right answers." This section describes each of these pieces for the collections used in the TREC-6 main tasks.

Table 2: Document collection statistics. Words are strings of alphanumeric characters. No stop words were removed and no stemming was performed.

	Size (megabytes)	# Docs	Median # Words/Doc	Mean # Words/Doc
Disk 1				
Wall Street Journal, 1987–1989	267	98,732	245	434.0
Associated Press newswire, 1989	254	84,678	446	473.9
Computer Selects articles, Ziff-Davis	242	75,180	200	473.0
Federal Register, 1989	260	25,960	391	1315.9
abstracts of U.S. DOE publications	184	226,087	111	120.4
Disk 2				
Wall Street Journal, 1990–1992 (WSJ)	242	74,520	301	508.4
Associated Press newswire (1988) (AP)	237	79,919	438	468.7
Computer Selects articles, Ziff-Davis (ZIFF)	175	56,920	182	451.9
Federal Register (1988) (FR88)	209	19,860	396	1378.1
Disk 3				
San Jose Mercury News, 1991	287	90,257	379	453.0
Associated Press newswire, 1990	237	78,321	451	478.4
Computer Selects articles, Ziff-Davis	345	161,021	122	295.4
U.S. patents, 1993	243	6,711	4445	5391.0
Disk 4				
the *Financial Times*, 1991–1994 (FT)	564	210,158	316	412.7
Federal Register, 1994 (FR94)	395	55,630	588	644.7
Congressional Record, 1993 (CR)	235	27,922	288	1373.5
Disk 5				
Foreign Broadcast Information Service (FBIS)	470	130,471	322	543.6
the *LA Times*	475	131,896	351	526.5
Routing Test Data				
Foreign Broadcast Information Service (FBIS)	490	120,653	348	581.3

3.1 Documents

TREC documents are distributed on CD-ROM's with approximately 1 GB of text on each, compressed to fit. For TREC-6, Disks 1–4 were all available as training material (see Table 2) and Disks 4 and new Disk 5 were used for the ad hoc task. Additional new FBIS data (also shown in Table 2) were used for testing in the routing task.

Documents are tagged using SGML to allow easy parsing (see fig. 2). The documents in the different datasets have been tagged with identical major structures, but they have different minor structures. The philosophy in the formatting at NIST has been to preserve as much of the original structure as possible, while providing enough consistency to allow simple decoding of the data. Both as part of the philosophy of leaving the data as close to the original as possible, and because it is impossible to check all the data manually, many "errors" remain in the data. The error-checking done at NIST has concentrated on allowing readability of the data rather than on correcting content. This means that there have been automated checks for control characters, special symbols, foreign language characters, for correct matching of the begin and end document tags, and for complete "DOCNO" fields (the field that gives the unique TREC identifier for the document). The types of "errors" remaining include fragment sentences, strange formatting around tables or other "non-textual" items, misspellings, etc.

The data on disk 5 and the FBIS routing test data are new TREC document sets (although the FBIS data on disk 5 was used as routing test data in TREC-5). The Foreign Broadcast Information Service provides (English translations of) selected non-U.S. broadcast and print publications. The documents on disk 5 were mostly from the early 1990's, and those used in the routing test data were mostly from the mid 1990's. The documents were provided

```
<DOC>
<DOCNO>FT911-3</DOCNO>
<PROFILE>AN-BEOA7AAIFT</PROFILE>
<DATE>910514
</DATE>
<HEADLINE>
FT 14 MAY 91 / International Company News:  Contigas plans DM900m east German
project
</HEADLINE>
<BYLINE>
By DAVID GOODHART
</BYLINE>
<DATELINE>
BONN
</DATELINE>
<TEXT>
CONTIGAS, the German gas group 81 per cent owned by the utility Bayernwerk, said
yesterday that it intends to invest DM900m (Dollars 522m) in the next four years
to build a new gas distribution system in the east German state of Thuringia.  ...
</TEXT>
</DOC>
```

Figure 2: A document extract from the *Financial Times*.

for TREC use by the Foreign Broadcast Information Service. The *LA Times* documents are a sample of the articles that appeared in the newspaper in 1989 and 1990. The articles are used by permission of the LA Times and were obtained for TREC use by Lexis-Nexis.

3.2 Topics

In designing the TREC task, there was a conscious decision made to provide "user need" statements rather than more traditional queries. Two major issues were involved in this decision. First, there was a desire to allow a wide range of query construction methods by keeping the topic (the need statement) distinct from the query (the actual text submitted to the system). The second issue was the ability to increase the amount of information available about each topic, in particular to include with each topic a clear statement of what criteria make a document relevant.

The topics used in TREC-1 and TREC-2 (topics 1–150) were very detailed, containing multiple fields and lists of concepts related to the subject of the topics. The ad hoc topics used in TREC-3 (151–200) were much shorter and did not contain the complex structure of the earlier topics. Nonetheless, participants in TREC-3 felt that the topics were still too long compared with what users normally submit to operational retrieval systems. Therefore the TREC-4 topics (201–250) were made even shorter: a single field consisting of a one sentence description of the information need. Figure 3 gives a sample topic from each of these sets.

One of the conclusions reached in TREC-4 was that the much shorter topics caused both manual and automatic systems trouble, and that there were issues associated with using short topics in TREC that needed further investigation [3]. Accordingly, the TREC-5 ad hoc topics re-introduced the title and narrative fields, making the topics similar in format to the TREC-3 topics. TREC-6 topics used this same format. A sample TREC-6 topic is shown in Figure 4, while Table 3 summarizes the length of the topics as measured by number of words.

3.2.1 Building topic statements

Ad hoc topics have been constructed by the same person who performed the relevance assessments for that topic since TREC-3. For TREC-6, NIST introduced a new procedure for developing topics with the hope that the resulting topics would strike a good balance

```
<num> Number:  051

<dom> Domain:  International Economics

<title> Topic:  Airbus Subsidies

<desc> Description:
Document will discuss government assistance to Airbus Industrie, or mention a
trade dispute between Airbus and a U.S. aircraft producer over the issue of
subsidies.

<narr> Narrative:
A relevant document will cite or discuss assistance to Airbus Industrie by the
French, German, British or Spanish government(s), or will discuss a trade dispute
between Airbus or the European governments and a U.S. aircraft producer, most
likely Boeing Co.  or McDonnell Douglas Corp., or the U.S. government, over
federal subsidies to Airbus.

<con> Concept(s):
1.  Airbus Industrie
2.  European aircraft consortium, Messerschmitt-Boelkow-Blohm GmbH, British
Aerospace PLC, Aerospatiale, Construcciones Aeronauticas S.A.
3.  federal subsidies, government assistance, aid, loan, financing
4.  trade dispute, trade controversy, trade tension
5.  General Agreement on Tariffs and Trade (GATT) aircraft code
6.  Trade Policy Review Group (TPRG)
7.  complaint, objection
8.  retaliation, anti-dumping duty petition, countervailing duty petition,
sanctions
```

```
<num> Number:  168
<title> Topic:  Financing AMTRAK

<desc> Description:
A document will address the role of the Federal Government in financing the
operation of the National Railroad Transportation Corporation (AMTRAK).

<narr> Narrative:
A relevant document must provide information on the government's responsibility to
make AMTRAK an economically viable entity.  It could also discuss the
privatization of AMTRAK as an alternative to continuing government subsidies.
Documents comparing government subsidies given to air and bus transportation with
those provided to AMTRAK would also be relevant.
```

```
<num> Number:  207

<desc> What are the prospects of the Quebec separatists achieving independence
from the rest of Canada?
```

Figure 3: The evolution of TREC topic statements. Sample topic statement from TRECs 1 and 2 (top), TREC-3 (middle), and TREC-4 (bottom).

```
<num> Number:  312
<title> Hydroponics

<desc> Description:
Document will discuss the science of growing plants in water or some substance
other than soil.

<narr> Narrative:
A relevant document will contain specific information on the necessary nutrients,
experiments, types of substrates, and/or any other pertinent facts related to the
science of hydroponics.  Related information includes, but is not limited to, the
history of hydroponics, advantages over standard soil agricultural practices, or
the approach of suspending roots in a humid enclosure and spraying them
periodically with a nutrient solution to promote plant growth.
```

Figure 4: A sample TREC-6 topic.

Table 3: Topic length statistics by topic section. Lengths count number of tokens in topic statement including stop words.

	Min	Max	Mean
TREC-1 (51–100)	44	250	107.4
title	1	11	3.8
description	5	41	17.9
narrative	23	209	64.5
concepts	4	111	21.2
TREC-2 (101–150)	54	231	130.8
title	2	9	4.9
description	6	41	18.7
narrative	27	165	78.8
concepts	3	88	28.5
TREC-3 (151–200)	49	180	103.4
title	2	20	6.5
description	9	42	22.3
narrative	26	146	74.6
TREC-4 (201–250)	8	33	16.3
description	8	33	16.3
TREC-5 (251–300)	29	213	82.7
title	2	10	3.8
description	6	40	15.7
narrative	19	168	63.2
TREC-6 (301–350)	47	156	88.4
title	1	5	2.7
description	5	62	20.4
narrative	17	142	65.3

between topics as diagnostic tools (i.e., neither too difficult nor too easy) and topics as realistic user inquiries.

The assessors came to NIST with an initial topic statement. These statements were prepared at home, and were treated as a user's statement of the information he or she was seeking. The statements usually reflected some consideration regarding the subject areas likely to be covered in the target documents, but otherwise were a simple description of the needed information without regard to retrieval system capabilities or document collection peculiarities.

Using these initial topic statements, the assessors explored (a subset of)[1] the TREC-6 ad hoc collection using NIST's PRISE retrieval system. There were two aims of the collection exploration phase: estimating the number of relevant documents in the collection and evaluating whether the topic could be judged consistently in the assessment phase. The assessors formed an initial PRISE query and judged the top 25 documents for relevance. If the top 25 contained no relevant documents or more than 20 relevant documents, the topic was abandoned. If the top 25 contained more than 5 but fewer than 21 relevant documents, the assessor continued to judge 75 more documents for a total of 100 documents judged. Finally, if the top 25 contained at least 1 relevant document but no more than 5 relevant documents, the assessor invoked the relevance feedback mechanism in PRISE, and judged the top 100 documents

[1] The collection used in the exploration phase consisted of the documents in the *Financial Times*, *LA Times*, and FBIS subcollections only. That is, the *Federal Register* and *Congressional Record* subcollections were excluded.

in the feedback result set. The total number of relevant documents found and the assessor's opinion as to how difficult the topic was to judge consistently were recorded for each topic.

The assessors came to NIST with a total of 120 candidate topics. Of those, 20 were discarded because there were no relevant documents in the top 25, and 9 were discarded because there were more than 20 relevant documents in the top 25. NIST selected 50 of the remaining 91 candidate topics based on having a range of estimated number of relevant, balancing the load across assessors, and eliminating topics that were considered difficult to judge.

Each of the final 50 topic statements were then reviewed by the assessors and NIST staff to ensure that the Narrative field of the topic statement accurately reflected how the assessor would judge documents for relevance. By judging 100 documents in the exploration phase, the assessors were able to see many of the issues they would have to deal with when assessing participants' results. Approximately five topics' Narrative fields were modified during this review, usually by removing restrictive clauses. The review also ensured that the Title field of the topics would meet the needs of those interested in exploring very short queries. Using guidelines suggested by Mark Sanderson of Glasgow University, the assessors created titles that contained up to three words that best described the topic.

3.2.2 Predicting topic difficulty

Recall that one of the goals for the TREC-6 topics was that they be neither too difficult nor too easy so they would be useful as diagnostic tools. In practice, predicting the difficulty of a topic is quite challenging. As an experiment to see whether NIST staff members could predict the difficulty of a topic based simply on the topic statement, nine members of the Natural Language Processing and Information Retrieval Group at NIST (including the authors) predicted how difficult each ad hoc topic would be. These predictions were made before the relevance assessments were performed, so the true answer could not be known at the time of prediction. Each person divided the topics into disjoint sets of easy topics, middling topics, and hard topics.

Once the relevance assessments were available and the participants' runs evaluated, a hardness measure was computed for each topic. The hardness measure used was introduced in TREC-2 and explored further in the TREC-5 Overview [9]. The hardness score for a topic T is computed as

mean Prec(100) for T	if T has 100 or more relevant documents
mean R-Prec for T	otherwise

where Prec(100) is precision at rank 100 and R-Prec is precision at rank R when there are R relevant documents. The means were computed over all Category A ad hoc submissions, including both manual and automatic runs. To arrive at "hard," "middling," and "easy" classifications of the topics, the hardness scores were sorted and divisions were made based on gaps in the hardness scores. This resulted in 12 hard topics, 11 easy topics, and 27 middling topics. These classifications were considered to be "the truth."

The Pearson correlation coefficient was computed between each person's prediction and the truth, and between different predictions. The Pearson correlation coefficient is suitable for interval values (so the difference between hard and easy was treated as more significant than the difference between middling and easy or middling and hard), and takes on a value between -1 and 1 inclusive. A value of 1 indicates perfect agreement, a value of -1 perfect disagreement, and a value of 0 chance agreement. The largest correlation between a prediction and the truth was .257, and the largest correlation between any two predictions was .387. To set this in context, in TREC-5 the (Spearman) correlation between hardness and topic number (that is, two items that have no actual correlation) was computed as .20. Thus, essentially none of the NIST staff members agreed with the truth or with one another.

This lack of agreement illustrates how little is known about what makes a topic difficult in the context of a particular document collection. Without an understanding of the factors that make a topic difficult, it is not possible to create test collections that balance difficulty (the ideal for the "diagnostic tool" test collection goal). The lack of understanding also impedes retrieval effectiveness. The Query Track, a new track to be introduced in TREC-7, was created to address this need. Each participant in the Query Track will create several different versions of queries for existing TREC topics. All participants will then run all versions of the queries. The goal of the track is to create a large enough pool of queries such that it will be possible to investigate query-dependent retrieval strategies.

3.2.3 Runs using different topic fields

As in TREC-5, groups who performed automatic ad hoc runs were required to do at least one run using a short version of the topic, i.e., the Description field.

These runs are tagged as "short, automatic" runs in the results section. Automatic runs that used only the Title field are tagged as "title, automatic" runs, and automatic runs that used the entire topic are tagged as "long, automatic" runs. Manual runs had no length requirements, and are assumed to be based on the entire topic text. Unfortunately, NIST did not inform the assessors that the different pieces of the topics would be used differently when they created the topics, and this confounds the conclusions that can be drawn from runs using different topic lengths.

The assessors treated each topic as a single unit, and did not necessarily repeat themselves in the different parts. Thus, some Description fields do not contain "Title" words — words that were specifically chosen to represent the core meaning of the topic! Specifically, none of the title words occur in the description for 5 topics, at least one title word is missing from the description for 22 topics, and the description contains all of the title words for the remaining 23 topics. Given this construction of the topics, a valid comparison is title *vs.* description+title, not title *vs.* description. A more thorough discussion of the effect of using different topic sections is given in section 5.

3.3 Relevance assessments

Relevance judgments are of critical importance to a test collection. For each topic it is necessary to compile a list of relevant documents—as comprehensive a list as possible. All TRECs have used the pooling method [7] to assemble the relevance assessments. In this method a pool of possible relevant documents is created by taking a sample of documents selected by the various participating systems. This pool is then shown to the human assessors. The particular sampling method used in TREC is to take the top 100 documents retrieved in each submitted run for a given topic and merge them into the pool for assessment. This is a valid sampling technique since all the systems used ranked retrieval methods, with those documents most likely to be relevant returned first.

3.3.1 Overlap

The effect of pooling can be measured by examining the overlap of retrieved documents. Table 4 summarizes the amount of overlap in the ad hoc and routing pools for each of the six TRECs. The first column in the table gives the maximum possible size of the pool. Since the top 100 documents from each run are judged, this number is usually 100 times the number of runs used to form the pool. However, in TREC-6 there were 13 High Precision runs that contributed a maximum of 10 documents each to the pool. The second column shows the number of documents that were actually in the pool (i.e., the number of unique documents retrieved in the top 100 across all judged runs) averaged over the number of topics. The percentage given in that column is the size of the actual pool relative to the possible pool size. The final column gives the average number of relevant documents in the pool and the percentage of the actual pool that was relevant. Starting in TREC-4, various tracks also contributed documents to the ad hoc or routing pools. These are broken out in the appropriate rows within Table 4. The order of the tracks is significant in the table—a document retrieved in a track listed later is not counted for that track if the document was also retrieved by a track listed earlier.

The tremendous drop in the size of the ad hoc pool between TREC-5 and TREC-6 reflects the difference in the number of runs NIST was able to assess. In TREC-5, participants were allowed to submit two manual and two automatic ad hoc runs, and all submitted runs were judged. However, many more participants submitted runs in TREC-6 than in TREC-5 and the amount of time available for assessing was 2 weeks shorter due to scheduling around other IR activities. Thus only one ad hoc run per participant was judged in TREC-6. (Participants were allowed to submit up to three ad hoc runs. They ranked the runs in order of preference as to which runs should be judged first when submitting the results. NIST judged every group's first choice. An investigation of the size of the pools if everyone's second choice were also merged into the pools showed that the pools would be too large for the assessors to finish in the available time.)

Table 4 shows that the average number of relevant documents per topic continues to decrease over the years. NIST has deliberately chosen more tightly focused topics to better guarantee the completeness of the relevance assessments.

4 Evaluation

An important element of TREC is to provide a common evaluation forum. Standard recall/precision figures and some single evaluation measures have been calculated for each run and are shown in Appendix A. A detailed explanation of the measures is also included in the appendix.

Additional data about each system was collected that describes system features and system tim-

Table 4: Overlap of submitted results

Ad Hoc				Routing			
	Possible	Actual	Relevant		Possible	Actual	Relevant
TREC-1	3300	1279 (39%)	277 (22%)	TREC-1	2200	1067 (49%)	371 (35%)
TREC-2	4000	1106 (28%)	210 (19%)	TREC-2	4000	1466 (37%)	210 (14%)
TREC-3	2700	1005 (37%)	146 (15%)	TREC-3	2300	703 (31%)	146 (21%)
TREC-4	7300	1711 (24%)	130 (08%)	TREC-4	3800	957 (25%)	132 (14%)
ad hoc	4000	1345	115	routing	2600	930	131
confusion	900	205	0	filtering	1200	27	1
dbmerge	800	77	2				
interactive	1600	84	13				
TREC-5	10,100	2671 (27%)	110 (04%)	TREC-5	3100	955 (31%)	113 (12%)
ad hoc	7700	2310	104	routing	2200	854	94
dbmerge	600	72	2	filtering	900	100	19
NLP	1800	289	3				
TREC-6	3,430	1445 (42%)	92 (06%)	TREC-6	4400	1306 (30%)	146 (11%)
ad hoc	3100	1326	89	routing	3400	979	105
NLP	200	113	2	filtering	1000	327	41
HP	130	6	1				

ing, and allows some primitive comparison of the amount of effort needed to produce the corresponding retrieval results. Due to the size of these system descriptions, they are not included in the printed version of the proceedings. The system descriptions are available on the TREC web site (http://trec.nist.gov).

5 Retrieval Results

One of the important goals of the TREC conferences is that the participating groups freely devise their own experiments within the TREC task. For some groups this means doing the routing and/or ad hoc task with the goal of achieving high retrieval effectiveness performance. For other groups, however, the goals are more diverse and may mean experiments in efficiency or unusual ways of using the data.

This overview of the results discusses the effectiveness of the systems and analyzes some of the similarities and differences in the approaches that were taken. In all cases, readers are referred to the system papers in this proceedings for more details.

5.1 TREC-6 ad hoc results

The TREC-6 ad hoc evaluation used new topics (topics 301–350) against two disks of training documents (disks 4 and 5). A dominant feature of the ad hoc task was the desire of groups to continue to work with both the short and long versions of the topics (as in TREC-5), and in addition to try a very short (title only) version. All three parts of the topics were built by the assessors, with the title being constrained to three words. Systems doing automatic query building were required to submit at least one run using the short version of the topic (only the description field), but in addition they could submit runs using either the very short (title) version or the long (full topic) version. Groups doing manual query building were assumed to be using the full topic.

There were 79 sets of official results for ad hoc evaluation in TREC-6, with 74 of them based on runs for the full (Category A) data set. Of these, 57 used automatic construction of queries, with 12 official very short runs, 29 short runs, and 16 long runs. Seventeen groups used manual construction. There were only five Category B runs from two groups.

5.1.1 Long (full topic) automatic runs

Figure 5 shows the recall/precision curves for the eight TREC-6 groups with the highest non-interpolated average precision using automatic construction of queries for the long (full topic) version of the topics (see Appendix A of this volume for definitions of the evaluation metrics). The runs are ranked by average precision and only one run is shown per group. These graphs (and others in this section) are not intended to show specific comparison of results

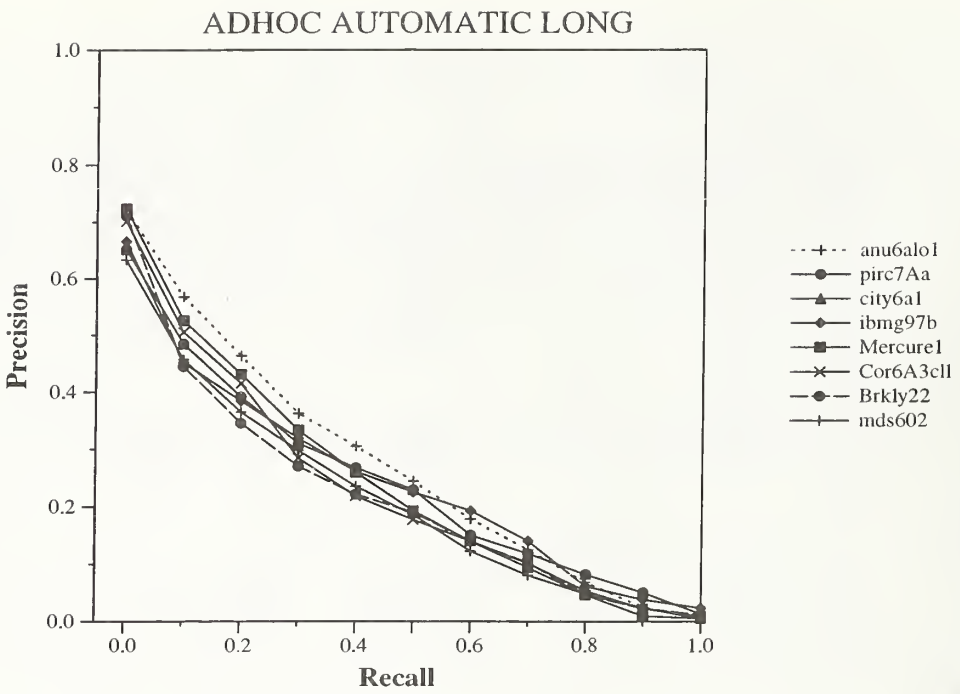


Figure 5: Recall/Precision graph for the top eight automatic ad hoc runs using the full topic.

across sites but rather to provide a focal point for discussion of methodologies used in TREC. For more details on the various runs and procedures, please see the cited papers in this proceedings.

anu6alo1 – Australian National University ("ANU / ACSys TREC-6 Experiments" by David Hawking, Paul Thistlewaite, and Nick Craswell) used a parallel architecture with an emphasis on efficiency. Improvements for TREC-6 include the use of the Cornell variant of the Okapi BM25 term weighting and major experiments to determine correct parameters for pseudo-relevance feedback (automatic relevance feedback using the top retrieved documents). These experiments included the use of "hot spots" in the top 20 documents for locating expansion terms and the use of the Robertson formula for term selection. The hot spots were defined as contiguous passages of text within a specified $p = 500$ characters of topic terms or phrases.

pirc7Aa – Queens College, CUNY ("TREC-6 English and Chinese Retrieval Experiments using PIRCS" by K.L. Kwok, L. Grunfeld, and J.H. Xu) used their spreading activation model for a two-stage search (initial search for doing pseudo-relevance feedback and a final search including expansion terms). They continued to work with 550-word subdocuments rather than dealing with multi-length documents, and generated the final score of a document as a weighted average of the scores of its three highest ranked subdocuments.

city6a1 – City University, London ("Okapi at TREC-6: automatic ad hoc, VLC, routing and filtering" by S. Walker, S.E. Robertson, and M. Boughanem) ran many experiments investigating the various parameters in the BM25 weighting technique, including adding provisions for using nonrelevant documents. Additionally a new formula for selecting expansion terms that considers 500 nonrelevant documents was tried. Note that the availability of many parameters for tuning allows the City group to systematically adjust their runs to specific functions. The first stage run (to get the expansion terms) was done as a high precision run; the final run was done with parameters appropriate to the length of the topic section being used. Expansion for the full topics selected the top 30 terms from the top 15 documents, multiplying weights in the original topic terms by 2.5 before doing the final retrieval. No phrases or pairs were used for TREC-6 (only single terms), however passages of between 4 and 30 paragraphs were used for the final runs only (not the initial runs for term expansion).

ibmg97b – IBM T.J. Watson Research Center ("The GURU System in TREC-6" by E. Brown and H. Chong) ran a probabilistic system that includes the use of lexical affinities (statistical phrases) in the topic. Through a series of experiments they found that performance was fairly insensitive to the distance between terms (up to a distance of 5 words was tested), but was very sensitive to the weighting of those terms. The best weight they found for these phrases was about 10% of that for the single terms in the topic. Note that no expansion was used in this run, and that only the title and description section were used as input (not the full topic).

Mercure1 – MSI/IRIT/SIG/CERISS ("Mercure at trec6" by M. Boughanem and C. Soule-Dupuy) continued their work with a spreading activation model. For TREC-6 they incorporated the Okapi/SMART BM25 weighting algorithm. Parameters in this algorithm were first set to achieve a high precision in the initial search to gather information for query expansion (similar to the City technique). Negative feedback using 500 low-ranked documents was also used in query expansion.

Cor6A3cl – Cornell/SaBIR Research ("Using Clustering and SuperConcepts Within SMART: TREC 6" by Chris Buckley, Mandar Mitra, Janet Walz, and Claire Cardie) concentrated on better initial retrieval and improved expansion. Many (unsuccessful) experiments were tried with phrases and Boolean filters, but were not used in the final run. The run for the full topics performed a clustering of candidate documents for topic expansion to help improve term selection. Note that this run did not use the title (by mistake), and the inclusion of the title gives an additional 13% improvement.

Brkly22 – University of California, Berkeley ("Phrase Discovery for English and Cross-language Retrieval at TREC-6" by Fredric C. Gey and Aitao Chen) used a probabilistic system involving heavy use of logistic regression. In TREC-6 they decided to try a new method for identifying phrases based on a mutual information measure that had been very successful in Chinese retrieval. The addition of phrases to their retrieval terms meant that the log-odds formula that they have used since TREC-2 needed to be modified to deal with the different patterns of occurrence associated with phrases as opposed to single terms. The use of phrases did not improve results in English over those that could be obtained from single terms.

mds602 – MDS/RMIT ("MDS TREC6 Report" by M. Fuller, M. Kaszkiel, C. Ng, P. Vines, R. Wilkinson, and J. Zobel) did a comprehensive factor analysis of various known successful components of retrieval, including stopwords, stemming, passage retrieval, term expansion, methods of combining results, and query length. This particular run combined four different sets of results: a baseline run, a run using the best 30 documents for expansion, a run using the best 150-word passages for expansion, and finally a run using the best 150-word passages from an already expanded query for additional expansion.

5.1.2 Short (description only) automatic runs

The method used at NIST to construct the topics for TREC-6 (discussed in sec. 3.2.3) caused very unusual results for the required short runs. The titles of the topics generally contained excellent topic descriptors, but for over half the topics some of these terms were not included in the description section of the topic. For many of the topics, therefore, the input to the short run consisted of a poor set of terms. Results from the "short" runs using only the description section should be viewed with great caution, therefore, and most groups redid their short runs to include the title (see individual papers for results).

However, as a way of continuing the discussion of the ad hoc results, results from the top eight short runs are shown in Figure 6.

city6ad – City University, London ("Okapi at TREC-6: automatic ad hoc, VLC, routing and filtering" by S. Walker, S.E. Robertson, and M. Boughanem) used the same methods as for the "long" run, but with different parameter settings (particularly applying less weight to negative feedback terms). Only the top ten documents were used for expansion, with 30 new terms being added.

LNaShort – Lexis-Nexis ("Query Processing in TREC-6)" by A. Lu, E. Meier, A. Rao, D. Miller, and D. Pliske) used their EUREKA toolbox to perform investigations in topic expansion and data fusion. Modified versions of two different search algorithms (Cornell's cosine Lnu.ltu and the Okapi BM25) were used along with three different methods of topic expansion

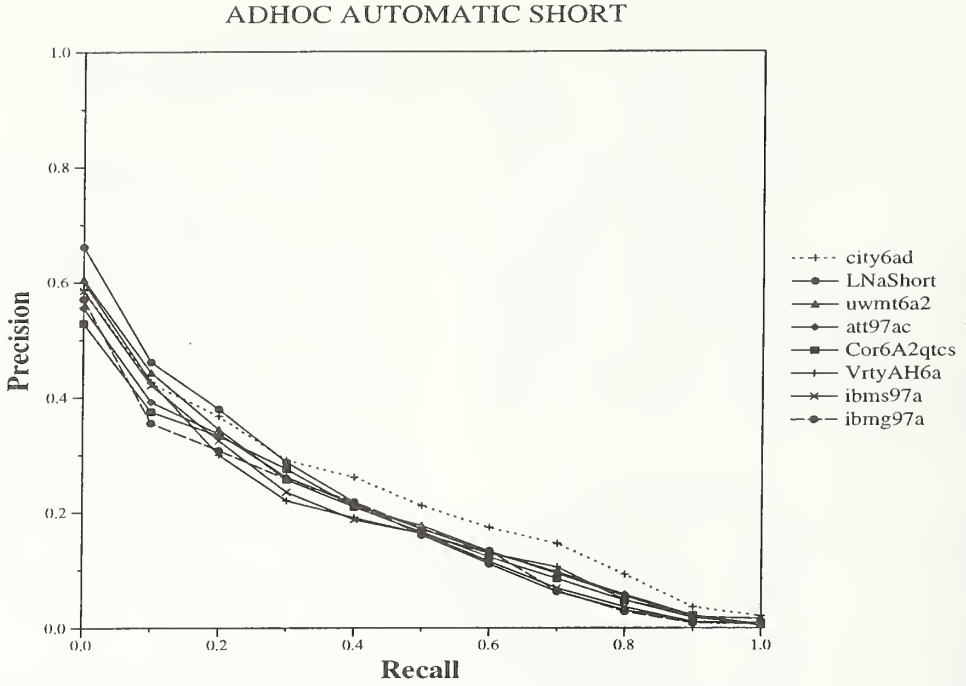


Figure 6: Recall/Precision graph for the top eight automatic ad hoc runs using the description only.

(WordNet, a Lexis-Nexis thesaurus, and Rocchio feedback) in a complex set of experiments involving merging results at several points in the process. This particular run involved first a merge of results from three runs using two weighting algorithms and WordNet or the thesaurus. The results of this were piped into a Rocchio feedback process, and then a final merge made of this output and the output of the first merging process.

uwmt6a2 – University of Waterloo ("Passage-Based Refinement (MultiText Experiments for TREC-6)" by G. Cormack, C. Clarke, C. Palmer, and S. To) made their first automatic runs in TREC-6. All the Waterloo runs used passage retrieval, with no collection-wide statistics, as the system is built for distributed architectures. The core technique for the short runs was a cover density method, which uses coordination-level matching for terms, followed by a secondary ranking using shortest substrings. The cover density technique was used to make the initial search to locate appropriate passages (of maximum length of 64 words) for use in expansion. To incorporate expansion, a modified implementation of the Okapi measure was used in the final search.

att97ac – AT&T Labs Research ("AT&T at TREC-6" by A. Singhal) is an outgrowth of the basic Cornell ad hoc approach. Two new techniques were tried in TREC-6. The first was the use of negative feedback in the Rocchio formula, based on documents ranked (in the initial search) at ranks 501–1000. This improved results from 3% to 4%. More improvement (6%–7%) was gained from a new method of reweighting the topic terms and reranking the top 50 documents prior to location of expansion terms. The top 20 documents were used for this expansion, adding 25 new terms and five phrases.

Cor6A2qtcs – Cornell/SaBIR Research ("Using Clustering and SuperConcepts Within SMART: TREC 6" by Chris Buckley, Mandar Mitra, Janet Walz, and Claire Cardie) tried a new method called "SuperConcepts" in the short run. The idea here was to divide the expansion terms into sets clustered around the initial topic terms, and adjust their weights, with the goal of producing a more balanced query that makes maximal use of the expansion terms without skewing the query. These SuperConcepts were then used for matching against the documents rather than using an expanded set of topic terms.

VrtyAH6a – Verity, Inc. ("Verity at TREC 6: Out-

of-the-Box and Beyond" by J. Pedersen, C. Silverstein, and C. Vogt) ran a series of experiments using several tools from the Verity system. Their baseline system used a variation of tf.idf weighting, but in addition they used a commercial shallow parser to find phrases and parts of speech. They used the Verity summarizer for both length normalization and as a method of finding expansion terms for relevance feedback (5 new terms added from the top 20 documents). They also used the Verity clustering tool to help decide whether to use feedback for a given topic, based on the distribution of the top 20 documents in 5 clusters from the top 1000 documents.

ibms97a – IBM T.J. Watson Research Center ("TREC-6 Ad-Hoc Retrieval" by M. Franz and S. Roukos) used a multi-pass strategy with a combination of unigrams (single terms) and bigrams (defined as order-dependent two-word phrases). The Okapi scoring algorithm was used with different parameter settings for the unigrams and bigrams, and the scores linearly combined in the first pass. The top 40 documents from this pass were used to find expansion unigrams and bigrams, which were then used in a second pass. The final pass used expansion terms from the second pass, but combined the scores with those from the first two passes.

ibmg97a – IBM T.J. Watson Research Center ("The GURU System in TREC-6" by E. Brown and H. Chong) used the same methods as for the long run, but took only the description as input.

5.1.3 Very short (title only) automatic runs

Figure 7 shows the recall/precision curves for the eight TREC-6 groups with the highest non-interpolated average precision using automatic construction of queries for the very short (title only) version of the topics.

city6at – City University, London ("Okapi at TREC-6: automatic ad hoc, VLC, routing and filtering" by S. Walker, S.E. Robertson, and M. Boughanem) used the same methods as for the long run, but with different parameter settings (particularly, applying less weight to negative feedback terms and no weighting for query term frequency). Only the top seven documents were used for expansion, with 20 new terms being added. Weights for the original topic terms were multipled by 3.5 instead of 2.5.

pirc7Aat – Queens College, CUNY ("TREC-6 English and Chinese Retrieval Experiments using PIRCS" by K.L. Kwok, L. Grunfeld and J.H. Xu) used the same system as for the long run. However for this title version they tried (without success) an experiment in document reranking before topic expansion that used selected topic term pairs from the description.

aiatB1 – Apple Computing. No paper was submitted for this run, so nothing is known about how it was made.

uwmt6a1 – University of Waterloo ("Passage-Based Refinement (MultiText Experiments for TREC-6)" by G. Cormack, C. Clarke, C. Palmer and S. To) This run was similar to their short topic run, but the final ranked list was based on a merging of three runs: a cover density run, a run using a modified Okapi weighting and third run using a modified Okapi expansion method. The expansion used 24 new terms.

Mercure3 – MSI/IRIT/SIG/CERISS ("Mercure at trec6" by M. Boughanem and C. Soule-Dupuy) did a similar run to their long run, but took only the title as input.

att97as – AT&T Labs Research ("AT&T at TREC-6" by A. Singhal) did a similar run to their short run, but took only the title as input.

LNaVryShort – Lexis-Nexis ("Query Processing in TREC-6)" by A. Lu, E. Meier, A. Rao, D. Miller, and D. Pliske) used their EUREKA toolbox in a simplified version of their description-only run. In this run only the Okapi algorithm was used, and 26 terms were added from the internally built thesaurus before a relevance feedback process (not Rocchio) was used to produce the final results.

iss97vs – Institute for Systems Science ("Concrete Queries in Specialized Domains: Known Item as Feedback for Query Formulation" by M. Leong) ran their major experiments in a manual mode. This run represents their baseline and was simply a query automatically constructed from the title. No query expansion was done.

The INQUERY system from the University of Massachusetts ("INQUERY Does Battle with TREC-6" by J. Allan, J. Callan, W.B. Croft, L. Ballesteros, D. Byrd, R. Swan, and J. Xu) did not make the above charts since they ran with only one-half the data (by mistake). Correct runs (see the paper) show that

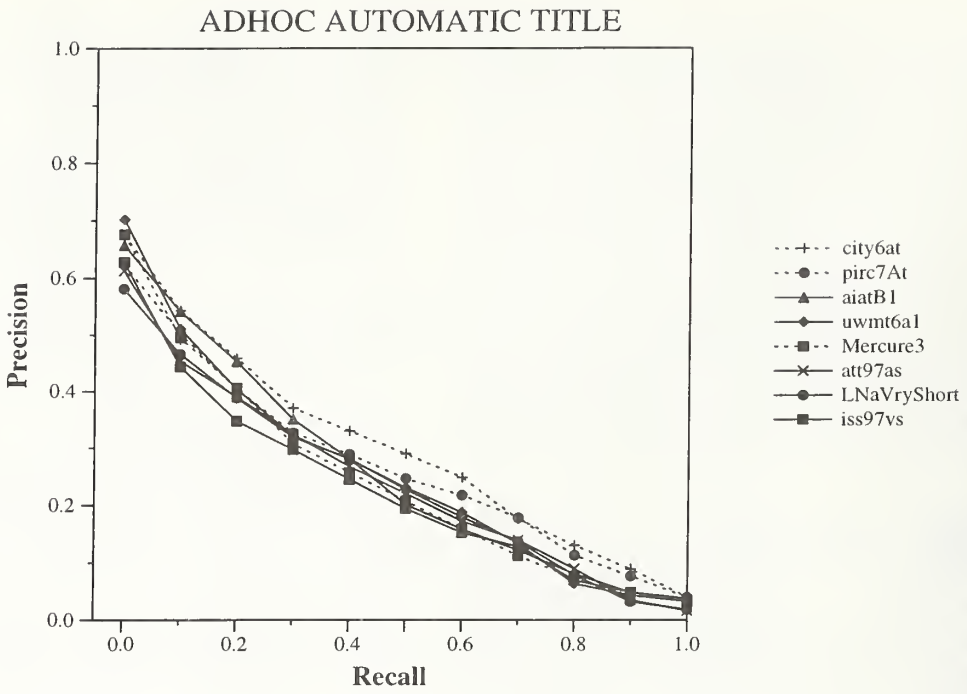


Figure 7: Recall/Precision graph for the top eight automatic ad hoc runs using the title only.

they performed as well as most of the systems shown. The basic retrieval model used in this system is a probabilistic belief network using weighting similar to the Okapi/SMART weighting, but employing complex query structures generated automatically. For TREC-6 they ran experiments in building phrase tables to help in phrase identification for input to that query structure.

5.1.4 TREC-6 ad hoc manual results

Figure 8 shows the recall/precision curves for the eight TREC-6 groups with the highest non-interpolated average precision using manual construction of queries. Note that manual query construction included user interaction in TREC-6; i.e., the rules allowed initial results to be viewed and the queries changed, with no restrictions on how much time could be spent. Therefore the amount of human effort required for these various techniques should be considered when comparing the retrieval results. A short summary of the techniques used in these runs follows; for more details on the various runs and procedures, see the cited papers in this proceedings.

uwmt6a0 – University of Waterloo ("Passage-Based Refinement (MultiText Experiments for TREC-6)" by G. Cormack, C. Clarke, C. Palmer, and S. To) used TREC-6 as an opportunity for experimentation on the correlation between the amount of user interaction and performance. The interfaces built for TREC-5 allowed extensive interaction with the system, and an average of 2.1 hours per topic was spent. Note that no actual ideal query was constructed during this time; the ranked list submitted to NIST was simply a list of all the documents that the searchers thought were relevant to the topic. This was done as part of an experiment in new ways of building test collections [1] rather than an investigation into manual query building.

CLAUG – CLARITECH Corp. ("Experiments in Query Optimization: The CLARIT System TREC-6 Report" by Natasa Milic-Frayling, Chengxiang Zhai, Xiang Tong, Peter Jansen, and David A. Evans) tested two different variations of relevance feedback. The searchers spent an average of about 20 minutes per topic and were constrained to constructing the initial manual query, modifying (adding/deleting/reweighting) terms based on inspecting documents, modifying Boolean query constraints (if used at all) and making relevance judgments. The CLAUG run represents a second pass using automatic pseudo-relevance feedback (from the top 50 documents) on top of a first pass (CLREL) which used 50 positive and

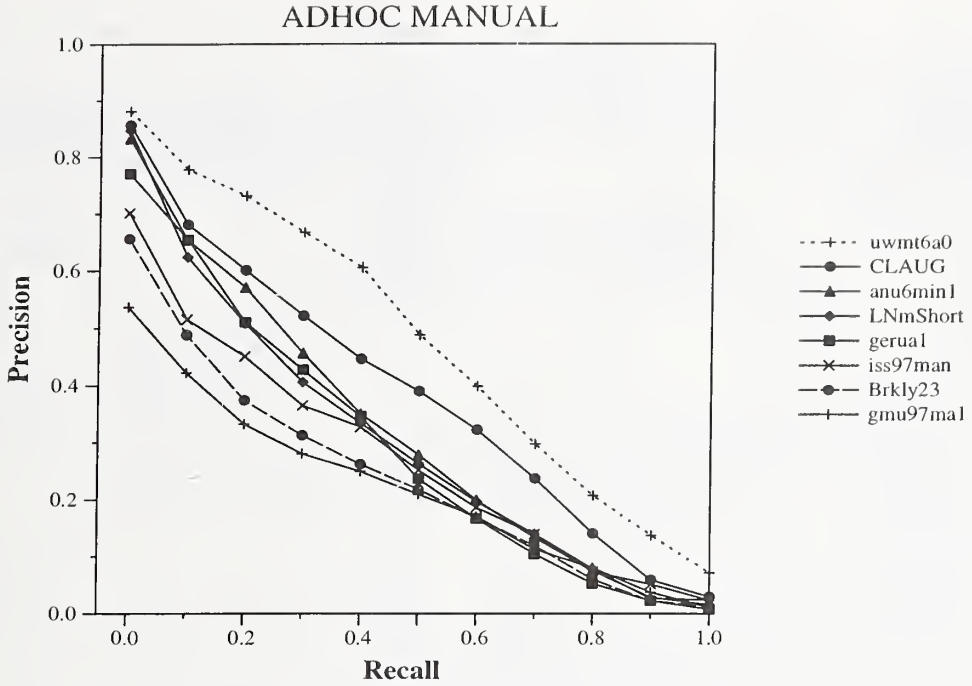


Figure 8: Recall/Precision graph for the top eight manual ad hoc runs.

30 negative terms selected via probabilistic term selection from user-judged documents.

anu6min1 – Australian National University ("ANU/ ACSys TREC-6 Experiments" by David Hawking, Paul Thistlewaite, and Nick Craswell) performed experiments investigating how well a relatively naive user could perform a series of edits on automatically generated queries, including removing or adding obvious terms, combining terms into phrases, altering weights, and grouping terms into concepts. These edits added 14% to average precision performance, and the further use of interactive modification improved precision by an additional 12% (with minimal improvement in recall).

LNmShort – Lexis-Nexis ("Query Processing in TREC-6)" by A. Lu, E. Meier, A. Rao, D. Miller, and D. Pliske) also performed experiments in manual editing of automatically generated queries. Their modified version of the Okapi BM25 algorithm was used to rank documents from automatic queries, and the top 20 documents were read looking for additional useful terms. The editing of the queries involved adding additional terms, removing negated terms, and doubling the frequency of the original query terms. This edited query was then used as input to some of the same automatic experiments used in the automatic runs. The addition of the manually selected terms gave major improvements.

geruna1 – GE Corporate R&D/Rutgers University ("Natural Language Information Retrieval TREC-6 Report" by T. Strzalkowski, F. Lin and J. Perez-Carballo) continued their investigations into contributions of natural language processing. This particular run represents experiments with the automatic generation of query-related summaries for the top 30 documents retrieved by the original topic. Users then added these summaries to the query if they "captured some aspects of relevant documents." These manually-expanded queries were run through the natural language processing modules to generate the final results.

iss97man – Institute for Systems Science ("Concrete Queries in Specialized Domains: Known Item as Feedback for Query Formulation" by M. Leong) used TREC as an environment to perform a specific experiment in manual query building. Their hypothesis was that expert users would be able to use very precise search terms, and this was tested using a two-stage search. In the first stage the users were given 20 minutes to find one or more highly relevant documents. In the second

stage the users were given 10 minutes to build a query that would return one of these highly relevant documents within the top 10 documents in the ranked list. This query was then used as the input to the manual run.

Brkly23 – University of California, Berkeley ("Phrase Discovery for English and Cross-language Retrieval at TREC-6" by Fredric C. Gey and Aitao Chen) did a manual reformulation of their queries.

gmu87ma1 – George Mason University/OIT/NCR ("Expanding Relevance Feedback in the Relational Model" by C. Lundquist, D. Holmes, D. Grossman, and O. Frieder) used their relational database model information retrieval system to experiment with pre-defined concept lists combined in different ways. These concepts were generated by first running a manual query, and then using relevance feedback and term-term association lists to generate more potential terms. These terms were then manually grouped into concept lists.

5.1.5 Discussion of TREC-6 ad hoc results

Since a dominant feature of the TREC-6 ad hoc task was the use of three different versions (lengths) of the topic, it is interesting to note the somewhat unexpected effects of this. The results using the title only (Very Short version) were surprisingly good, whereas those that used only the description (Short version) were considerably worse. Results using all three parts of the topic (Full version) were approximately the same as the results using the title only. These effects were generally consistent across all participating groups.

However, it would be unwise to generalize these results by claiming that systems do as well with very short (three word) topics as with much longer ones. As with most information retrieval testing, there is a huge variation across topics. For example, the table below shows the number of topics for which a given length was better than the other two lengths as measured by average precision for two sets of runs, the

	Title	Short	Long
City	21	11	18
PIRCS	23	9	18

City University runs and the CUNY (PIRCS system) runs. The counts in the table show that each topic length had some topics for which it formed the best query.[2]

Many of the TREC-6 topics turned out to have very few relevant documents, and in most of these cases all of the relevant documents were retrievable using only the keywords in the title. In these cases the full topic simply adds more "noise" to the query. An extreme example of this is Topic 312 shown in Fig. 4 on page 8. The single title word, "hydroponics," appears in all of the 11 relevant documents and in 18 documents total. This simple separation between relevant and non-relevant documents is not true of all the TREC-6 topics, and is probably not true of user requests in general, but the highly precise terms in the TREC-6 titles both illustrate the power of a well-constructed user query and create biased results.

Some of the participating groups used the same retrieval techniques for all topic lengths. Given the above discussion, this is likely to be less effective than adapting techniques to the specific parts of the topic being used. For example, City University used fewer documents (top 7 *vs.* top 10 *vs.* top 15) for mining of expansion terms, added fewer expansion terms (20 *vs.* 30 *vs.* 30), and gave more weight to the original topic terms (3.5 *vs.* 2.5 *vs.* 2.5) for the long, short, and title versions of the topic. Lexis-Nexis, Cornell, and the University of Waterloo tried different techniques for different lengths of topics. Cornell used a new technique, SuperConcepts, for the description only runs and not the full topic runs. Lexis-Nexis used a much simpler version of their elaborate data fusion techniques for the title runs, whereas the University of Waterloo used more data fusion for their title-only run.

A second theme that dominated the TREC-6 ad hoc task was the continued spread of the newer, better techniques across most participating groups. Some techniques have now become standard usage, and TREC-6 saw both some interesting adaptations of these techniques to new retrieval models, and some further elaboration of these techniques by their originators. Table 5 shows some of these now-standard techniques, along with their spread and elaboration history.

Six different research areas are shown in the table, with research in many of these areas triggered by changes in the TREC evaluation environment. For example, the use of subdocuments or passages was caused by the initial difficulties in handling full text documents, particularly excessively long ones. The

[2]While the counts are very similar, the set of topics for which one length is better than the others differs between the two groups.

Table 5: Use of new techniques in the ad hoc task

TREC-2	TREC-3	TREC-4	TREC-5	TREC-6
baseline for most systems beginning of Okapi weighting experiments	Okapi perfects BM25 algorithm	new SMART weighting algorithm new INQUERY weighting algorithm	use of Okapi / SMART weighting algorithms by other groups	adaptations of Okapi / SMART algorithms in most systems
use of subdocuments by PIRCS system	heavy use of passages / subdocuments			use of passages in relevance feedback
	beginning of expansion using top X documents	heavy use of expansion using top X documents	beginning of more complex expansion schemes	more sophisticated expansion experiments by many groups
	beginning of manual expansion using other sources	major experiments in manual editing / user-in-the-loop	continued user-in-the-loop experiments	extensive user-in-the-loop experiments
	initial use of "data fusion"	continued use of "data fusion"	continued use of "data fusion"	more complex use of "data fusion"
			beginning of more concentration on initial topic	continued focus on initial topic, including title

use of better term weighting, including correct length normalization procedures, made this technique less used in TREC's 4 and 5, but it resurfaced in TREC-6 to facilitate better input to relevance feedback.

The table also shows the rapid spread of successful technology across groups. Most groups spent TREC-1 simply struggling to scale-up their systems from searching several megabytes of documents to searching 2 gigabytes of documents. However, two new techniques were already being used by TREC-2. The Okapi system from City University, London was compelled to experiment with new term weighting algorithms since their initial algorithm did not scale. By TREC-3 this algorithm had been "perfected" into the BM25 algorithm now in use by many of the systems in TREC-6. Continuing along this same row in table 5, two other systems (the SMART system from Cornell and the INQUERY system from the University of Massachusetts) changed their weighting algorithms in TREC-4 based on analysis comparing their old algorithms to the new BM25 algorithm. By TREC-5 many of the groups had adopted these new weighting algorithms, with the early adopters being those systems with similar structural models to the Okapi, SMART, or INQUERY systems.

TREC-6 saw even further expansion of the use of these new weighting algorithms (alternatively called the Okapi/SMART algorithm, or the Cornell implementation of the Okapi algorithm). In particular, many groups adapted these algorithms to new models, often involving considerable experimentation to find the correct fit. For example IRIT modified the Okapi algorithm to fit a spreading activation model, IBM modified it to deal with unigrams and trigrams, and the Australian National University and the University of Waterloo used it in conjunction with various types of proximity measures. Of major note is the fact that City University also ran major experiments with the BM25 weighting algorithm in TREC-6, including extensive exploration of the various existing parameters, and addition of some new ones!

The second new technique started back in TREC-2 (the second line of table 5) was the use of smaller sections of documents, called subdocuments, by the PIRCS system at City University of New York. Again this issue was forced by the difficulty of using the PIRCS spreading activation model for documents having a wide variety of lengths. By TREC-3 many

of the groups were also using subdocuments, or passages, to help with retrieval. But, as mentioned before, TREC's 4 and 5 saw far less use of this technique as many groups dropped the use of passages due to minimal added improvements in performance.

TREC-6 saw a revival in the use of passages, but generally only for specific uses. Whereas the PIRCS system continued to use 550-word subdocuments for all its processing, most systems used passages only in the topic expansion phase. The Australian National University worked with "hot spots" of 500 characters surrounding the original topic terms to locate new expansion terms. AT&T used overlapping windows of 50 words to help rerank the top 50 documents before selecting the final documents for use in expansion. The University of Waterloo used passages of maximum length 64 words to select expansion terms, whereas Verity used their automatic summarizer for this purpose. Two groups (Lexis-Nexis and MDS) performed major experiments in the use of passages, particularly when employed in conjunction with other methods as input to data fusion.

The query expansion techniques shown in the third and fourth lines of the table were started when the topics were substantially shortened in TREC-3. As described in section 3.2, the format of the topics was modified to remove a valuable source of keywords: the concept section. In the search for some technique that would automatically expand the topic, several groups revived an old technique of assuming that the top retrieved documents are relevant, and then using them in relevance feedback. This technique, which had not worked on smaller collections, turned out to work very well in the TREC environment.

By TREC-6 almost all groups were using variations on expanding queries using information from the top retrieved documents (pseudo-relevance feedback). There are many parameters needed for success here, and groups continue to investigate the best settings for these parameters. Whereas there is general system convergence on some of these parameters, such as how many top documents to use for mining terms, how many terms to select, and how to weight those terms, these still need to be tested by systems adopting these techniques. Additionally there continue to be elaborations on these techniques, such as the several groups (City University, AT&T, and IRIT) that successfully got information from negative feedback in TREC-6.

Groups that built their queries manually also looked into better query expansion techniques starting in TREC-3. By TREC-5 these had evolved into very extensive user-in-the-loop experiments. Many of the manual experiments seen in TREC-6, however, go back to the simpler scenario of having users edit the automatically-generated query, or having users select documents to be used in automatic relevance feedback. Several of the groups had specific user strategies that they tested in TREC-6.

Data fusion (line 5 in table 5) has been used in TREC by many groups in various ways, but has increased in complexity over the years. In TREC-6, for example, several groups such as Lexis-Nexis used multiple stages of data fusion, including merging results from different term weighting schemes and from different query expansion schemes.

The final major research area shown in this table started in TREC-5. This area is illustrated in the experiments by several groups to "mine" more information from the initial topic, rather than simply treating the topic as a bag of potential keywords for input to the system. The INQUERY system from the University of Massachusetts has worked in all TREC's to automatically build more structure into their queries, based on information they have mined from the topic. In an effort to further improve performance, more groups have experimented with other information in the initial topic. This includes making more use of term proximity features (Australian National University, University of Waterloo, and IBM), clustering potential query expansion terms to maintain the initial topic balance (Cornell University), and looking for clues that would suggest a need for more emphasis on certain topic terms (AT&T and CUNY).

5.2 TREC-6 routing results

The routing evaluation used a specifically selected subset of the training topics against a new set of test documents. The routing tests in TREC-4 and TREC-5 had serious mismatches in the training and the test data, and it was determined to try routing in TREC-6 using very similar training and testing data. To this end the topics were the TREC-5 topics that had reasonable numbers of relevant documents from the FBIS data. To replace the "bad" topics, nine new topics, with minimal training data, were created, bringing the total to 47 topics. The test data for TREC-6 was additional FBIS documents.

There was a total of 34 sets of results for routing evaluation, with 33 of them based on runs for the full data set. Of the 33 systems using the full data set, 28 used automatic construction of queries, and 5 used manual construction. The single Category B routing run used automatic construction of queries.

Figure 9 shows the recall/precision curves for

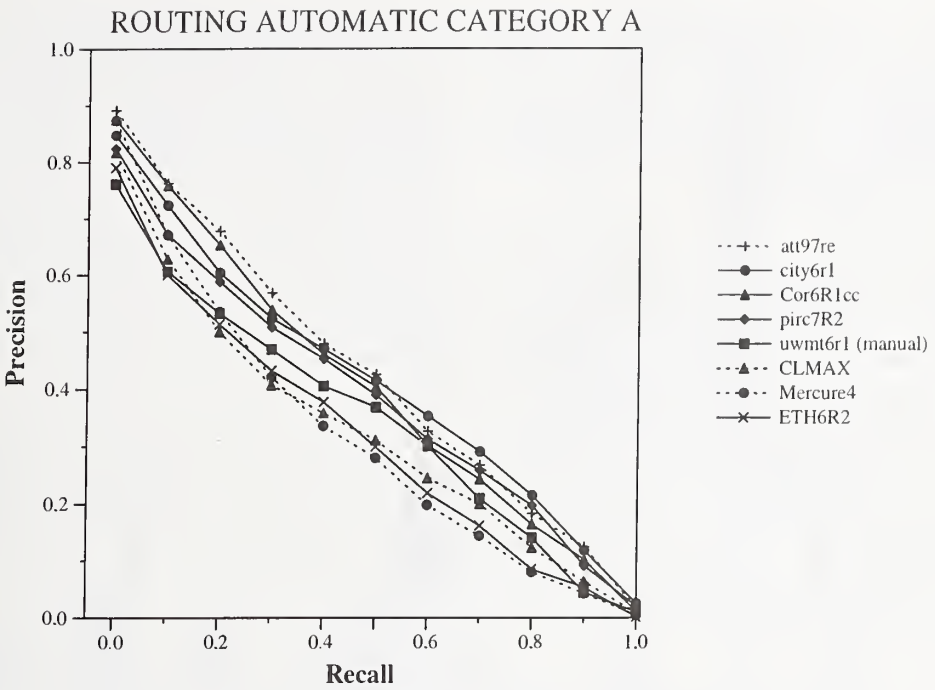


Figure 9: Recall/Precision graph for the top eight routing runs.

the eight TREC-6 groups with the highest non-interpolated average precision for the routing queries. The runs are ranked by the mean average precision over the 47 topics. A summary of the techniques used in these runs follows. For more details on the various runs and procedures, please see the cited papers in this proceedings.

att97re – AT&T Labs Research ("AT&T at TREC-6" by A. Singhal) used a variant of the Cornell TREC-5 routing algorithm. The modification added a version of the machine learning technique of boosting to the query refinement phase of the basic algorithm that includes the use of word pairs, DFO optimization, and query zones. The boosting added a small advantage (approximately 4%) compared to the algorithm without boosting.

city6a1 – City University, London ("Okapi at TREC-6: Automatic ad hoc, VLC, routing, filtering and QSDR" by S. Walker, S.E. Robertson, and M. Boughanem) explored iterative methods of term weighting with a major goal of avoiding overfitting the training data. This run is the result of merging 24 queries generated by picking various numbers of terms from the training set. For half the queries, the full FBIS training set was used; for the other half the training set was split in half, and one part was used to pick the terms and the other part was used to weight the terms.

Cor6A3cl – Cornell/SaBIR Research ("Using Clustering and SuperConcepts Within SMART: TREC 6" by C. Buckley, M. Mitra, J. Walz, and C. Cardie) added the SuperConcept technique to their basic TREC-5 routing algorithm. The SuperConcept technique attempts to maintain a balance between the different concepts represented in the original query by having expansion terms related to a particular concept of the original query share the total weight allocated to the concept. This technique did not improve the routing results as compared to the basic TREC-5 routing algorithm. However, the DFO optimization had not been modified to work with SuperConcept weighting, so improvements may still be possible.

pirc7R2 – Queens College, CUNY ("TREC-6 English and Chinese Retrieval Experiments using PIRCS" by K.L. Kwok, L. Grunfeld, and J.H. Xu) continued experimentation with merging of results from multiple runs. Five runs using different retrieval methods were used: one run using the topic only, one run using the training data only (only FBIS documents), two runs using

the genetic algorithms from TREC-5, and one using a new back propagation algorithm. This run combined the results of a combination of the first four methods with the back propagation run. This combined result was superior to all of the component runs.

uwmt6r1 – University of Waterloo ("Passage-Based Refinement (MultiText Experiments for TREC-6)" by G. Cormack, C Clarke, C. Palmer, and S. To) submitted a manual run using tiered Boolean queries that were refined interactively. An initial manual query was decomposed into basic components and combinations of these components were assigned to tiers such that combinations that retrieved relevant documents occurred in early tiers. The refinement produced small, but consistent, improvements over the original queries, and a future goal is to automate the process.

CLMAX – Claritech Corporation ("Experiments in Query Optimization: The CLARIT System TREC-6 Report" by N. Milic-Frayling, C. Zhai, X. Tong, P. Jansen, and D.A. Evans) explored the benefits of using different term selection methods in different parts of the query refinement process. For this run, they developed different queries using different term selection strategies and then, for each topic, selected the query that performed the best on the training data. They discovered that the query that performed best on the training data was not always the query that performed best on the test data: the results of the *CLMAX* run were not better than some of the component runs, while the results of the combined run using the actual best-performing queries were significantly more effective than each of the component runs.

Mercure1 – MSI/IRIT/SIG/CERISS ("Mercure at trec6" by M. Boughanem and C. Soulé-Dupuy) continued their work with a spreading activation model. The initial queries were automatically built from the topics and then expanded using the top 30 terms from relevance backpropagation. To prevent the query from becoming too much like the already retrieved relevant documents, terms that occurred in relevant documents that were not retrieved in the top 1000 by this system were given a small extra weight.

ETH6R2 – Swiss Federal Institute of Technology (ETH) ("ETH TREC-6: Routing, Chinese, Cross-Language and Spoken Document Retrieval" by B. Mateev, E .Munteanu, P. Sheridan, M. Wechsler, and P. Schäuble) ran further experiments with the U-measure. The top 300 single-word features and top 300 phrases were selected based on this measure. These features were then grouped using a similarity thesaurus and used as one component of a combined run. The other components consisted of a straight Lnu.ltn query expansion run and a run using feature co-occurrence matrices.

The best mean average precision for a routing run in TREC-5 was .386 (using 39 topics) and for TREC-6 was .420. While this is a 9% improvement, a greater improvement was generally expected. As stated earlier, the test data in the TREC-4 and TREC-5 routing tasks were not very similar to the training data, whereas the TREC-6 task was designed to use a homogeneous data set. Indeed, the histogram given in Figure 10 shows that the training and test data do have similar numbers of relevant documents for most topics.

At this point, it is unclear why the routing results are not better than they are. It is possible that while the numbers of relevant documents in the training and test set are comparable, the relevant documents in each set don't "look like" each other. However, this is unlikely since both sets of documents come from a common source. Another hypothesis suggested by Amit Singhal [5] is that the relevance judgments are less consistent for routing than they are for the ad hoc task. Since some routing topics have been used many times, and therefore have relevance judgments spanning many years, the judgments *are* likely to be less consistent than for the ad hoc task. On the one hand, a ceiling of .42 in retrieval effectiveness because of relevance judgment inconsistency is extremely unlikely. On the other hand, the techniques used to create the routing queries from the training data may magnify the effects of inconsistent judgments. It may be instructive to explore the stability of the routing techniques in the face of different relevance judgments, especially given that real user judgments are known to be extremely volatile [4].

The routing guidelines allowed participants to use any/all of the relevance judgments available for a topic in the training for that topic. The filtering track, in contrast, specified that training could only be done with previous FBIS judgments. Some groups ran routing experiments comparing the results from the two different training sets, and reached contradictory conclusions regarding which was better. For example, the Daimler Benz group concluded that using

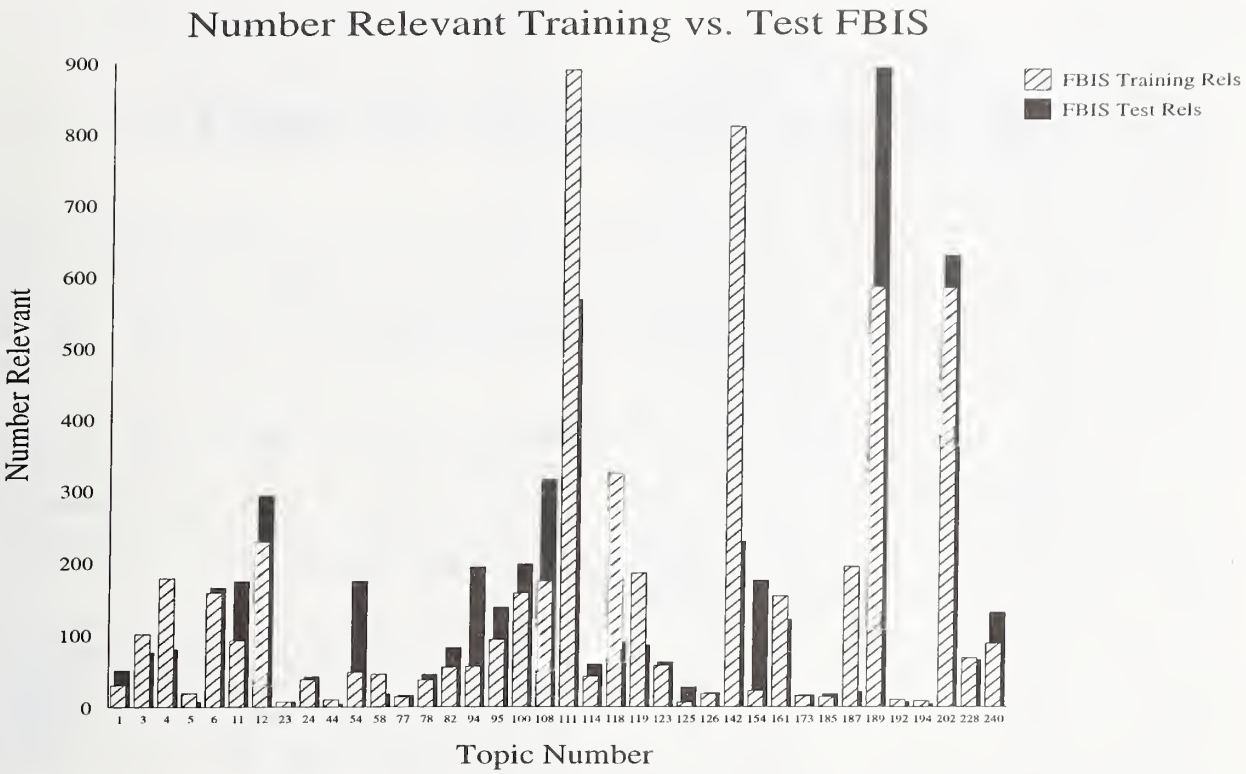


Figure 10: Comparison of the number of relevant documents in the training and test FBIS collections.

all of the training examples made the training set too unlike the test set for their classifier and using only the FBIS examples would be better. CISRO concluded precisely the opposite: that the FBIS training examples were too limiting and the variety introduced by judgments on other sources improved their results. Verity suggested a compromise of using all the judgments while emphasizing a particular source. The optimum trade-off between specificity and generality of the training data is clearly different for different techniques, and should be explored further.

6 Summary

TREC continues to grow both in number of participants and in number of tasks. The main tasks provide an entry point for new participants and provide a baseline of retrieval performance; the tracks invigorate TREC by introducing research in new areas of information retrieval. The Chinese track and the earlier Spanish track were the first (large-scale) formal tests of retrieval systems for languages other than English. The new Cross-Language Track exploits the current high interest in cross-language retrieval and serves as a testing platform both in the United States and Europe. The Spoken Document Retrieval Track, another track introduced in TREC-6, has joined the speech recognition and information retrieval communities, providing opportunities for rich interaction.

As always, it is difficult to summarize the many retrieval experiments that were performed in the context of TREC-6. Each group ran multiple experiments that resulted in their TREC submission, and readers are urged to explore the individual papers in this proceedings. In addition, Appendix B, "Summary Performance Comparisons TREC-2, TREC-3, TREC-4, TREC-5, TREC-6" by Karen Sparck Jones presents a snapshot of various system performances, particularly in the high precision end of the retrieval spectrum.

Several general conclusions can nevertheless be drawn from the main task experiments. The routing results suggest that there is still much to be learned about the stability of methods used to construct routing queries. The surprisingly good performance of the very short (titles only) ad hoc runs demonstrates the power of a few well-chosen query words—just as the relatively poor performance of the short ad hoc runs demonstrates how important it is to include those words. While this difference between the very short and short versions of the topics confounds results, there are suggestions that changing retrieval strate-

gies according to query length is beneficial.

The final session of each TREC workshop is a planning session for future TRECs. One of the tasks in this year's session was to contain the growth of TREC tasks in the face of finite resources at NIST to support TREC. Accordingly, the routing task was retired as a main task, though it will continue as a sub-task of the filtering track in TREC-7. The decision to retire the routing task was based on both the general agreement that the filtering task is a more realistic routing-type problem than the routing task as it has been defined in TREC, and that routing research can continue with the six routing collections that have already been built. Two tracks, NLP and Chinese, have also been discontinued for TREC-7, while a new Query Track will be introduced in TREC-7. The Query Track is designed to foster research on the effects of query variability on retrieval performance by creating and distributing many different queries derived from existing TREC topics.

Acknowledgments

The authors gratefully acknowledge the continued support of the TREC conferences by the Intelligent Systems Office of the Defense Advanced Research Projects Agency. Thanks also go to the TREC program committee and the staff at NIST. The TREC tracks could not happen without the efforts of the track coordinators; our special thanks to them.

References

[1] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98)*, 1998. To appear.

[2] Donna Harman. Analysis of data from the second Text REtrieval Conference (TREC-2). In *Proceedings of RIAO94*, pages 699–709, 1994.

[3] Donna Harman. Overview of the fourth Text REtrieval Conference (TREC-4). In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 1–23, October 1996. NIST Special Publication 500-236.

[4] Linda Schamber. Relevance and information behavior. *Annual Review of Information Science and Technology*, 29:3–48, 1994.

[5] Amit Singhal. AT&T at TREC-6. In *Proceedings of TREC-6*, 1998. In this volume.

[6] K. Sparck Jones. Reflections on TREC. *Information Processing and Management*, 31(3):291–314, 1995.

[7] K. Sparck Jones and C. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

[8] Jean Tague-Sutcliffe and James Blustein. A statistical analysis of the TREC-3 data. In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3) [Proceedings of TREC-3.]*, pages 385–398, April 1995. NIST Special Publication 500-225.

[9] Ellen Voorhees and Donna Harman. Overview of the fifth Text REtrieval Conference (TREC-5). In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, pages 1–28, November 1997. NIST Special Publication 500-238.

Chinese Document Retrieval at TREC-6

Ross Wilkinson
Mathematical and Information Sciences, CSIRO
Email: Ross.Wilkinson@cmis.csiro.au

January 9, 1998

1 Multilingual Document Retrieval in TREC

The TREC-6 conference was the fourth year in which document retrieval in a language other than English was carried out. In TREC-3, 4 groups participated in an ad hoc retrieval task on a collection of 208 Mbytes of Mexican newspaper text in the Spanish language. In TREC-4 there were 10 groups who participated, once again in an ad hoc document retrieval task on the same Mexican newspaper texts but with new topics. In TREC-5 there was a change of document corpus and new topics for the Spanish ad hoc retrieval task and a corpus of documents and topics to support ad hoc retrieval in the Chinese language was introduced for the first time. In TREC-6 there was two tracks in which languages other than English were explored. In the Chinese track, a second set of topics were evaluated against the existing corpus. In the cross-lingual track experiments were conducted where queries in one language were used against a document corpus in another language. This report concentrates solely on the Chinese track.

2 Chinese Language

In the Chinese language each character represents at least a complete syllable, rather than a letter as in other languages. Many characters are also single syllable words. The total number of characters is therefore quite large and somewhat ill-defined. A literate adult would typically recognise at least 5-6,000 characters. The various modern standards define between 10-12,000 characters, although if early and ancient literature is included the number rises to approximately 100,000. Chinese is agglutinating – there is no space between consecutive characters, except perhaps, at the end of a sentence.

Thus to perform retrieval in Chinese, the basis has to be characters unless the text is pre-segmented into words. Segmentation is difficult – not even humans will always agree on correct segmentation, and there has been much research in successful segmentation of Chinese [1].

3 Retrieval Task

The retrieval task for the Chinese track is exactly the same as the standard ad-hoc task in TREC. A given database of texts and a fixed set of topics are supplied. The task is to return a ranked list of 1,000 documents for each of the topics. For each topic, at least one run using only the description part of the topic was encouraged. The topics were supplied in both English and Chinese. Either the English could be used so that cross lingual retrieval could be explored, or the language of the document collection could be used for monolingual experiments.

A 164,811 document collection including documents from both the People's Daily and the Xinhua News Agency was used. There was no segmentation information supplied. It was 170 Megabytes as raw text. There were 26 new topics constructed. There were on average 114 relevant docs per topic for Chinese.

4 Chinese Results

The 12 groups who took part in TREC-6 Chinese generally explored the use of words vs. n-grams and methods of manually modifying queries. Some work was also done on retrieval methods particularly appropriate to Chinese retrieval. We summarize these approaches before discussing their comparative retrieval effectiveness.

City University: The experiments at City University used the Okapi system for their Chinese retrieval experiments. The tried both a character based retrieval and a word retrieval. Words were discovered using a greedy algorithm using a 70,000 word dictionary. With both character and word approaches, the use of phrases were explored. A number of probabilistic relationships were investigated based on the relative probability of a phrase appearing given that both constituents have appeared.

Claritech Corporation: The Claritech used the Clarit system for bi-gram character retrieval, and then applying automatic feedback. A comparison was made between long and short queries, as well as using manual intervention. Manual intervention of about 7 minutes per query produced a 5% gain. Feedback was routinely helpful.

Cornell University: Cornell again approached Chinese retrieval with no Chinese expertise but a very good retrieval system – the SMART system. They approached the task by using character based retrieval augmented with character bi-grams. They applied standard expansion techniques, as well as SuperConcepts. The expansion on single characters no longer gave a performance gain.

Information Technology Institute: The Information Technology Institute applied a novel matching algorithm for character based retrieval using positional information. They then combined the technique with expansion terms selected from 3-grams. This gave a gain. Further filtering of terms was not helpful.

Institute of Systems Science: The Institute of Systems Science carried out only automatic runs by combining both bi-gram approaches and word based approaches. These gave gains. They then investigated the nature of n-grams concluding that 2-grams are roughly equivalent to function words, 3-grams are roughly equivalent to names, and 4-grams are roughly equivalent to concepts.

MDS, RMIT: The RMIT approach was to combine several automatic runs based upon characters, bi-grams, and words found by dictionary methods. Two dictionary methods were used, based on dictionary and mutual information, with the dictionary giving superior results. The best results were obtained by combining bi-grams and words, and combining with expansion terms as well.

Queens College, CUNY: Last year, Queens had the most successful automatic approach based on identification and indexing on short-words, words of no more than 3 characters. This approach was again applied successfully using a 43K lexicon. However given the success of bi-grams, a combination approach was taken this year by combining the results of short-words and bi-grams. Results again were very good.

Swiss Federal Institute of Technology: ETH used fully automatic techniques to index using bi-grams. They concentrated on reducing the size of the vocabulary by using a stop list to partially segment. The achieved a large reduction in the corresponding dictionary. A manual approach was used using feedback. 40-50 minutes were spent on each topic, including feedback.

University of California, Berkeley: In TREC-5, the Berkeley group put a lot of effort into building a good dictionary of 140,000 words to use to automatically segment the text. They noticed that this dictionary still did not contain many important indexing terms, in particular names, so they developed further techniques for identifying these names and further augmented their dictionary with 10,000 entries. In their runs they used a word based indexing approach which was then either manually or automatically augmented with additional query terms.

University of Massachusetts, Amherst: The UMass approach to TREC-6 was exactly the same as with TREC-5. A hidden Markov model was used to segment text in the University of Massachusetts approach. The resulting queries used characters, groups of characters, and words.

Experiments were then conducted using the Local Context Analysis approach to term expansion.

University of Montreal: The Montreal effort concentrated on improved word identification algorithms by using more sophisticated morphological analysis. The results of this approach was then compared to the bi-gram approach. The word based approach gave slightly better performance, but the authors believe that more gain is possible after more effective segmentation.

University of Waterloo: The University of Waterloo used individual character indexing augmented by phrases based on adjacent characters using their Multitext approach. Topics were constructed manually by interacting with the collection. Several queries were constructed per topic and results merged manually. 1.5–2 hours were used to formulate queries per topic. The results gave the best manual run.

General Remarks

This was the second year of retrieving Chinese documents. Retrieval effectiveness was again very high. Median performance was above 0.5. With the average of the best run for any group at about 0.7, and the best single group performance at 0.62, it is clearly difficult to distinguish between approaches. What is less clear is why results are so high – it could be the data, the queries or the nature of Chinese. Until a new collection with different queries are obtained this question remains open. What is clear is that given the very high level of performance it is very hard to obtain insights into differences that individual techniques might make.

Team	Better run
City	character better (5%)
Claritech	character only
Cornell	character only
ITI	character only
ISS	character better (18%)
MDS	word better(1%), combination best
Queens	word better (4%), combination best
ETH	character only
Berkeley	words only
UMass	character better (2%)
Montreal	words better (1%)
Waterloo	user selected

Table 1: Comparing character-based and word-based approaches

Despite this and following last year's results, it is still the case that bi-gram approaches are comparable with any other individual technique and have the advantage of not requiring the difficult task of segmentation in order to employ word-based approaches (see Table 1). It has been suggested by several participants that this is due to the greater semantic content of characters compared to any sub-word element in English and other European languages. It was again the case that most groups obtained some gain from expansion and that manual intervention helped.

References

[1] R. Sproat and C.L. Shih. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351, 1990.

Cross-Language Information Retrieval (CLIR) Track Overview

Peter Schäuble, Páraic Sheridan

Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich, Switzerland

1 Introduction

Cross-Language Information Retrieval (CLIR) was a new task in the TREC-6 evaluation. In contrast to the *multilingual* track included in previous TREC evaluations, which was concerned with information retrieval in Spanish or Chinese, the *cross-language* retrieval track focuses on the retrieval situation where the documents are written in a language which is *different* than the language used to specify the queries. The TREC-6 track used documents in English, French and German and queries in English, French, German, Spanish and Dutch.

There are many applications or scenarios in which a user of a retrieval system may be interested in finding information written in a language other than the user's native or preferred language. In some applications, a user may want to discover all possible relevant information in a multilingual textbase, irrespective of the language of the relevant information. This may be the case when searching certain collections of legal or patent information for example. In other cases a user may even have some language comprehension ability in the languages of the documents (passive vocabulary) but may not have a sufficiently rich active vocabulary in the document languages to confidently specify queries in those languages. In this case a cross-language search which permits the user to specify native language queries but retrieves documents in their original language is useful. Cross-language retrieval also has the added advantage of requiring only one query to a multi-lingual text collection, rather than having a user submit individual queries in each of the languages of interest.

Situations where a retrieval system user is faced with the task of querying a multilingual document collection are becoming increasingly common. These range across document collections made up of documents from local offices of multinational companies, collections composed of documents from different regions of multilingual countries such as Switzerland or Canada, or the document

collections of large organisations such as the United Nations or European Commission. It is however, the global information infrastructure of the internet that has been largely responsible for the growing awareness of a need for cross-language information retrieval systems. This has in turn led to a growing body of research into the problems of cross-language retrieval and the development of several different approaches for CLIR.

1.1 Approaches to CLIR

Central to the problem of cross-language information retrieval is the matching of user queries specified in one language against documents written in a different language - *crossing the language boundary.* This may be achieved by translating the user queries, translating the documents, or by translating both the queries and documents, perhaps into some intermediary or interlingual representation. Both queries and documents might be translated into a common controlled indexing vocabulary for example, either manually or automatically. More common however, are approaches which work with the full text of documents and queries for matching. These can usefully be classified according to what resources are used to aid in crossing the language boundary; machine translation, machine-readable dictionaries, or corpus resources.

NEC in Japan (Yamabana et al., 1998) have used machine translation technology for cross-language retrieval by translating users' queries in an interactive process using both dictionaries and statistical information derived from bilingual corpora. The machine translation company Systran have also reported work addressing the cross-language retrieval problem with a system that includes machine translation technology at all stages of the retrieval process and which allows users to include detailed linguistic information in queries (Gachot et al., 1998). Researchers at Carnegie Mellon University (CMU) (Carbonell et al., 1997) have investigated the use of translation techniques from the study of Example-based Machine Translation (EBMT) for cross-language retrieval. The evaluation of machine translation as an approach to CLIR has been facilitated by a collaboration between the University of Maryland and the LOGOS corporation in the TREC-6 CLIR track.

Approaches to cross-language information retrieval which rely on corpus resources include the use of Latent Semantic Indexing (LSI) by researchers at Bellcore and elsewhere (Rehder et al., 1998) (Littman et al., 1998) (Landauer and Littman, 1991), the Generalised Vector Space Model proposed by CMU (Carbonell et al., 1997), a corpus-based approach from CNR in Pisa (Peters and Picchi, 1997), and our work at ETH which uses *similarity thesauri* for query translation (Mateev et al., 1996) (Sheridan et al., 1997) (Sheridan and Schäuble, 1997) (Sheridan and Ballerini, 1996). A common thread in these approaches is the use of corpus resources to train the cross-language retrieval mechanism or to build the information structures used for retrieval. Several of these approaches have also been evaluated in this TREC-6 track.

The use of existing linguistic resources, especially machine-readable bilingual dictionaries, is a natural approach to cross-language retrieval. Researchers at the Xerox Research Centre Europe (Hull and Grefenstette, 1996) (Grefenstette et al., 1998), the University of Massachusetts (Ballesteros and Croft, 1996) (Ballesteros and Croft, 1998), and the Computing Research Laboratory (CRL) of New Mexico State University (NMSU) (Davis, 1998) (Davis and Ogden, 1997), have extensively investigated the use of machine-readable dictionaries for cross-language retrieval and have variously addressed ways of overcoming some of the problems with dictionary-based translation. Many groups have submitted results based on machine-readable dictionaries for the TREC-6 evaluation.

Apart from the use of controlled vocabulary indexing or the use of manually-constructed multilingual thesauri, the TREC-6 CLIR evaluation has covered all classes of approaches to cross-language retrieval.

2 CLIR-Track Task Description

The Cross-Language Information Retrieval (CLIR) track requires the retrieval of either English, German or French documents that are relevant to topics formated in different languages. Participating groups were to choose any cross-language combination, for example English queries against German documents or French queries against English documents. In order have a baseline retrieval performance measurement for each group, the results of a monolingual retrieval experiment in the document language were also to be submitted, corresponding to each cross-language experiment run. For instance, if a cross-language experiment was run with English queries retrieving German documents then the result of the equivalent experiment where German queries retrieve German documents must also be submitted. These results are considered comparable since the queries are equivalent across the languages.

The document collections for the CLIR track were not however equivalent in English, French and German. The different document collections used in each language are outlined in Table 2. The Associated Press collection consists of newswire stories in English. The French SDA collection is a similar collection of newswire stories from the Swiss news agency (Schweizerische Depeschen Agentur). The German document collection has two parts. The first part is composed of further newswire stories from the Swiss SDA while the second part consists of newspaper articles from a Swiss newspaper, the 'Neue Zuercher Zeitung' (NZZ). The newswire collections in English, French and German were chosen to overlap in timeframe (1988 to 1990) for two reasons. First, since a single set of topics was going to be formulated to cover all three document languages, having the same timeframe for newswire stories increased the likelihood of finding a greater number of relevant documents in all languages. The second reason for the overlapping timeframe was to allow groups who use corpus-based approaches

to cross-language retrieval to investigate what useful corpus information they could extract from the document collections being used.

Document Collections			
Doc. Language	**Source**	**No. Documents**	**Size**
English	AP news, 1988-1990	242,918	760MB
German	SDA news, 1988-1990	185,099	330MB
	NZZ articles, 1994	66,741	200MB
French	SDA news, 1988-1990	141,656	250MB

Table 1: Document Collections used in the CLIR track.

The use of corpus-based approaches was further facilitated by the comparable nature of the SDA collections in German and French. These are newswire stories prepared by the same Swiss news agency, though the stories do not overlap perfectly and are not translated between the two languages. The stories are produced independently in each language, but there is in fact a high overlap of stories (e.g. international events) which are of interest in both the German-speaking and French-speaking parts of Switzerland. One of the resources provided to CLIR track participants was a list of 83,698 news documents in the French and German SDA collections which were likely to be comparable based on an alignment of stories using news descriptors assigned manually by the SDA reporters, the dates of the stories, and common cognates in the texts of the stories.

The 25 test topic (query) descriptions were provided by NIST in English, French and German. The 25 topics were equivalent across the languages. Participating groups who wished to test other query languages were permitted to create translations of the topics in their own language and use these in their tests, as long as the translated topics were made publicly available to the rest of the track participants. The final query set therefore also had translations of the 25 topics in Spanish, provided by the University of Massachusetts, and Dutch, provided by TNO in the Netherlands.

Although not strictly within the definition of the cross-language task, participation by groups who wanted to run mono-lingual retrieval experiments in either French or German using the CLIR data was also permitted. Since the CLIR track was run for the first time this year, this was intended to encourage new IR groups working with either German or French to participate. The participation of these groups also helped to ensure that there would be a sufficient number of different system submissions to provide the pool of results needed for relevance judgements.

The evaluation of CLIR track results was based on the standard TREC-adhoc evaluation measures. Participating groups were free to experiment with

different query length and with both automatic and manual experiments according to the definitions used for the main TREC adhoc task.

3 Results

A total of thirteen different groups, representing seven different countries, participated in the TREC-6 CLIR track. The first important result from this track therefore, was the participation of new groups in TREC, especially new groups from Europe. Participating groups were encouraged to run as many different experiments as possible, both with different kinds of approach to CLIR and with different language combinations. An overview of the submitted runs is given in Table 3. This shows that the main query languages were used equally, each used in 29 experiments, whereas English was somewhat more popular than German or French as the choice for the document language to be retrieved. This is in part because the groups who used the query translations in Spanish and Dutch only evaluated those queries against English documents. A total of 95 result sets were submitted for evaluation in the CLIR track.

Language Combinations						
	Query Language					
Doc. Language	*English*	*German*	*French*	*Spanish*	*Dutch*	**Total**
English	7	15	10	2	6	**40**
German	12	10	4	-	-	**26**
French	10	4	15	-	-	**29**
Total	**29**	**29**	**29**	**2**	6	**95**

Table 2: Overview of submissions to CLIR track.

An important contribution to the track was made by a collaboration between the University of Maryland and the LOGOS corporation, who provided a machine translation of German documents into English. Only the German SDA documents were prepared and translated in time for the submission deadline, but machine translation output of the NZZ document collection is now also available. This MT output was provided to all participants as a resource, and was used to support experiments run at ETH (Mateev et al., 1996), Duke University (Rehder et al., 1998), Cornell University (Buckley et al., 1998), Berkeley (Gey and Chen, 1998), and the University of Maryland (Oard and Hackett, 1998).

Cross-Language retrieval using dictionary resources was the approach taken in experiments submitted by groups at New Mexico State University (Davis and Ogden, 1998), University of Massachusetts (Allan et al., 1998), the Com-

missariat à l'Energie Atomique of France (Elkateb and Fluhr, 1998), the Xerox Research Centre Europe (Grefenstette et al., 1998), and at TNO in the Netherlands. Machine readable dictionaries were obtained from various sources, including the internet, for different combinations of languages, and used in different ways by the various groups.

Corpus-based approach to CLIR were evaluated by ETH, using similarity thesauri, and the collaborative group of Duke University, the University of Colorado, and Bellcore, who used latent semantic indexing (LSI). An innovative approach for cross-language retrieval between English and French was also tested at Cornell University. This approach was based on the assumption that there are many similar-looking words (near cognates) between English and French and that, with some simple matching rules, relevant documents could be found without a full translation of queries or documents.

The offer allowing groups to participate in a monolingual capacity using the document collections for French or German was accepted by groups at Dublin City University (Smeaton et al., 1998), University of Montreal (Nie and Chevallet, 1998), and IRIT France (Boughanem and Soulé-Dupuy, 1998).

An overview of results for each participating group are presented in Figure 1. This figure represents the results based on only 21 of the total 25 test queries. The remaining 4 queries have not yet been fully judged for relevant documents. The figure shows results for each group and each document language for which experiments were submitted. The y axis represents the average precision achieved over the set of 21 queries for the *best* experiment submitted by each group and each document language. Cross-language experiments are denoted by, for example, *'X to French'*, whereas the corresponding monolingual experiments are denoted, *'French'*. For example, the figure shows that the best experiment submitted by Cornell University performing cross-language retrieval of French documents achieved average precision of 0.2.

Note that the presentation of results in Figure 1 does not distinguish between fully automatic cross-language retrieval, and those groups who included some interactive aspect and user involvement in their experiments. The groups at Xerox, Berkeley and Dublin submitted experiments which involved manual interaction.

Although Figure 1 does not provide a sound basis for between-group comparisons, we can make some general comments on the overall results. Comparing cross-language results to the corresponding monolingual experiments, it seems that cross-language retrieval is performing in a range of roughly 50% to 75% of the equivalent monolingual case. This is consistent with previous evaluations of cross-language retrieval. Groups at ETH, Cornell, Xerox, TNO, Berkeley and Maryland have all achieved cross-language results in this range, compared to good levels of monolingual retrieval performance. While the LSI group, CEA and NMSU have also achieved good relative cross-language performance, their baseline monolingual retrieval performance is somewhat lower than the rest.

A slightly more detailed analysis of results can be achieved by plotting some

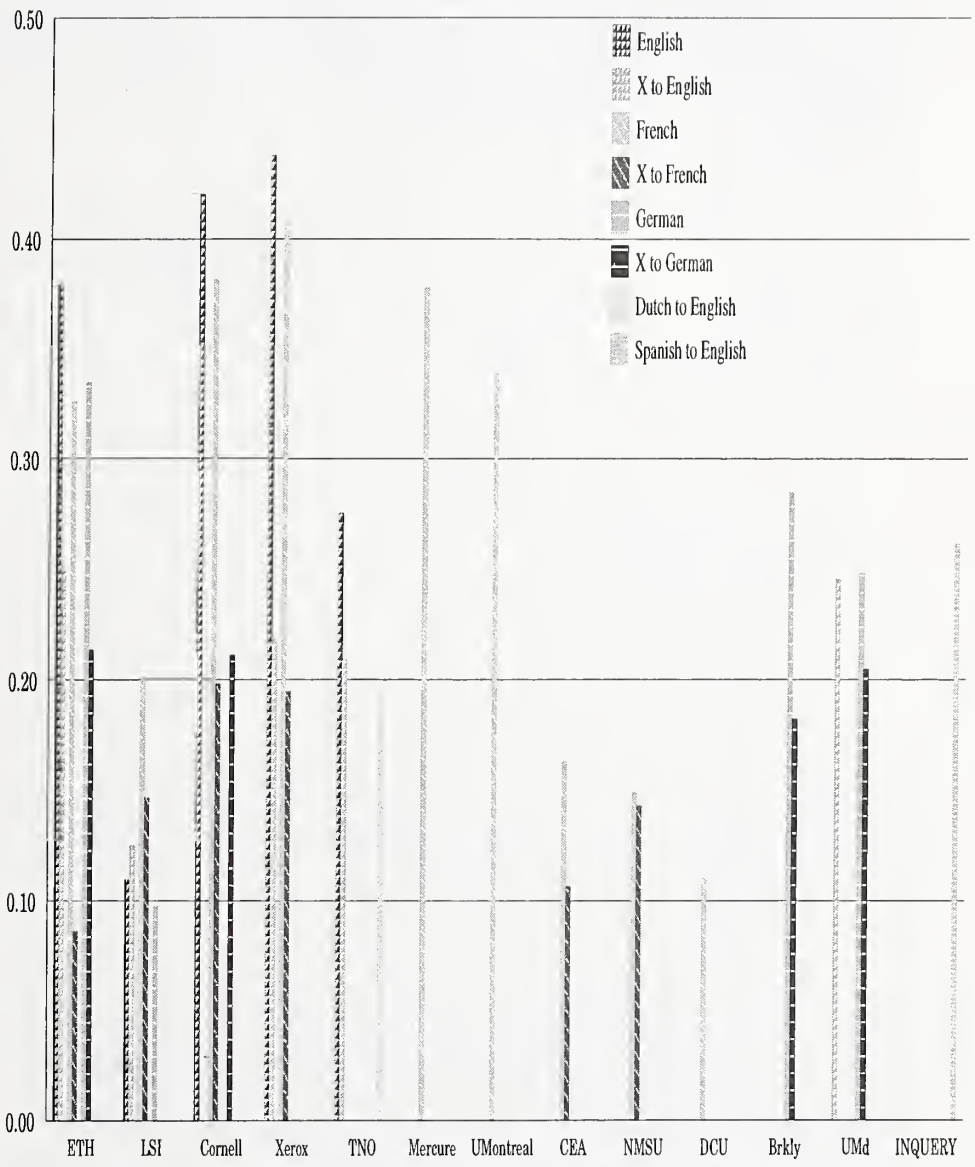


Figure 1: CLIR Track Results (Average Precision, best run)

of the experiments on standard recall/precision graphs, although the data for this is taken from an earlier point when only *thirteen* of the twenty five queries had been completely judged for relevance. We do not provide either a complete or detailed view of all the CLIR track results, but present in Figure 2 a selection of the results achieved for cross-language retrieval of French documents with English queries, and in Figure 3 some of the results of cross-language retrieval to German documents from English queries.

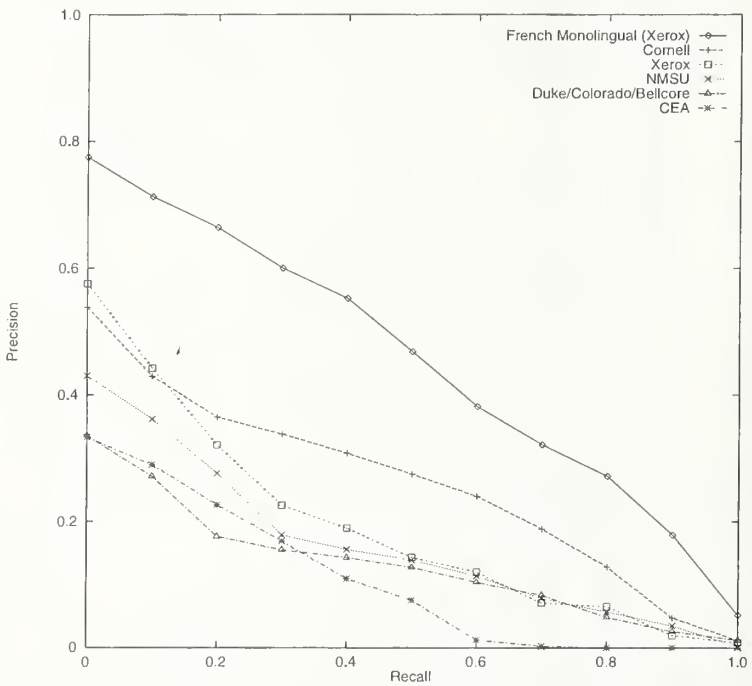


Figure 2: English-to-French CLIR: 13 Queries

The English-to-French CLIR experiments illustrated in Figure 2 include Cornell's cognate-matching approach, machine-readable dictionary approaches by Xerox, NMSU, and CEA, and the LSI run of the Duke/Colorado/Bellcore group which used corpus information derived from an alignment of the French SDA collection with English machine-translation output of the comparable German SDA collection. In each case we have presented the best automatic run submitted by these groups. The baseline for comparison is the monolingual French-French retrieval experiment submitted by the Xerox Research Centre Europe (Grenoble). The success of Cornell's approach is testament to the high degree of cognate overlap in English and French.

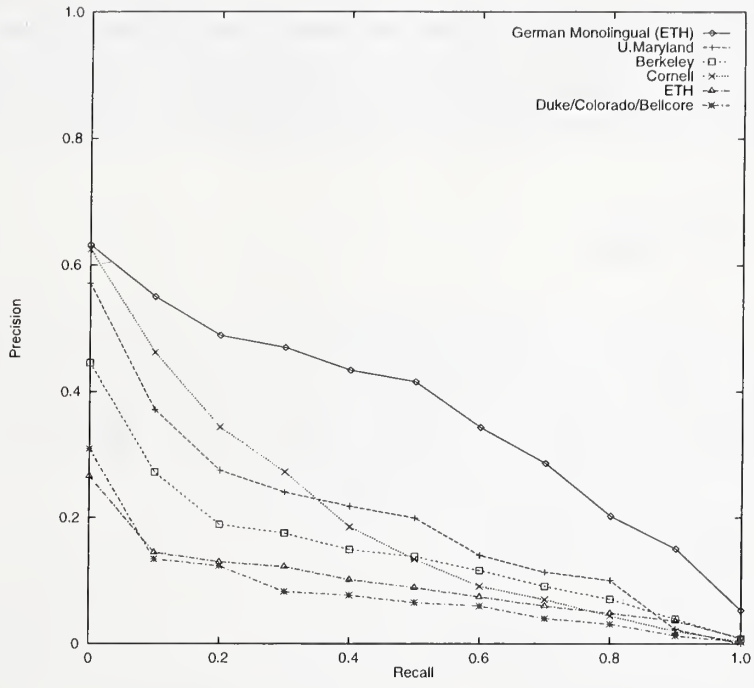


Figure 3: English-to-German CLIR: 13 Queries

Figure 3 includes English-to-German experiments using the LOGOS Machine Translation output (i.e. matching the English queries directly against the English MT output) from Maryland and Cornell, a dictionary-based experiment from Berkeley, and corpus-based approaches from ETH and the LSI group. The ETH experiment was based on a similarity thesaurus constructed using document *summaries* ('Title' and 'Lead' fields) of the German collection aligned with English MT output of the document summaries. The LSI approach was based on an alignment between documents and translations of the German SDA documents. The upper baseline in Figure 3 is provided by the monolingual German experiment run at ETH Zurich. Comparison of the cross-language results shows that for the 13 queries included, retrieval based on the MT output of the documents has done relatively well. The results also suggest that further investigation of the use of corpus material is necessary, though the corpus used in these experiments must be considered to be of poor quality with respect to the alignment of documents.

On the whole, the CLIR results are encouraging, especially given that this is the first year this track has been run. Ten groups submitted cross-language

experiments. Many different approaches to cross-language retrieval have been tried and evaluated, and groups using each of the different approaches have achieved good results. As the track grows in the future, the challenge will be to find a concise way of presenting and comparing the performance of the different groups with different approaches running experiments with different language combinations.

4 Outlook

The outlook for the cross-language retrieval track is good. We believe that this track has shown already this year that cross-language retrieval is feasible. Groups now have a foundation on which to develop their approaches and a baseline against which to compare future performance. The existence now of a substantial test collection with documents in three languages and parallel queries in five languages, together with the relevance judgements provided by NIST, is an important asset to the CLIR community.

The TREC test collection is just one resource for CLIR however. It seems from the results presented in the previous section that the availability of resources, whether they be machine translation, machine-readable dictionaries, or high-quality corpora, is a key determinant of the level of cross-language retrieval performance that can be achieved. The acquisition of better dictionary and corpus resources is likely to be an important source of improved CLIR performance in the future.

The CLIR track has again illustrated some of the problems of automatically translating natural language text long known in the field of machine translation. The importance of translating the correct senses of words (e.g. *'logging'*) and of correctly translating multi-word units (e.g. *'fast food'*) as a single entity rather than word-by-word has been illustrated in several of the test queries. Groups which have used automatic MT, either for document translation or query translation, have demonstrated however that MT has an important contribution to make to cross-language information retrieval.

Plans for the TREC-7 CLIR track include the acquisition of more document data in each of German, French and English. There are no plans to add new document languages in the immediate future, although this has been discussed as a long-term goal of the track. The formulation of topic descriptions for TREC-7 CLIR will be spread out over three different sites, with native French, German and English speakers responsible for the formulation of the topics in each language. The three sites, at NIST, Bonn, Germany, and Lausanne, Switzerland, will then also be responsible for completing the relevance judgements of documents in the retrieved pool. The involvement of these sites in the process is aimed at ensuring the fluency of topic descriptions and the speed and reliability of relevance judgements for our future test collections.

Acknowledgements: We would like to express our appreciation to the Neue Zuercher Zeitung (NZZ), the Schweizerische Depeschen Agentur (SDA), and the Associated Press (AP) for making their data available to the TREC community. The authors would also like to thank NIST, on behalf of the CLIR track participants, for their work in converting this data into a valuable CLIR test collection.

References

Allan, J., Callan, J., Croft, W. B., Ballesteros, L., Byrd, D., Swan, R., and Xu, J. (1998). INQUERY Does Battle with TREC-6. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Ballesteros, L. and Croft, W. B. (1996). Dictionary-based Methods for Cross-lingual Information Retrieval. In *Proceedings of the 7th International DEXA Conference on Database and Expert Systems Applications.*

Ballesteros, L. and Croft, W. B. (1998). Statistical methods for cross-language information retrieval. In Grefenstette, G., editor, *Cross-Language Information Retrieval*, chapter 3. Kluwer Academic Publishers, Boston.

Boughanem, M. and Soulé-Dupuy, C. (1998). Mercure at Trec-6. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Buckley, C., Mitra, M., Walz, J., and Cardie, C. (1998). Using Clustering and SuperConcepts Within SMART: TREC 6. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Carbonell, J., Yang, Y., Frederking, R., Brown, R. D., Geng, Y., and Lee, D. (1997). Translingual information retrieval: A comparative evaluation. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence.*

Davis, M. and Ogden, W. (1997). QUILT: Implementing a Large-Scale Cross-Language Text Retrieval System. In *Proceedings of the 20th ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, PA*, pages 27–31.

Davis, M. and Ogden, W. (1998). Free Resources and Advanced Alignment for Cross-Language Text Retrieval. In *Proceedings of the Sixth Text Re-*

trieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.

Davis, M. W. (1998). On the effective use of large parallel corpora in cross-language text retrieval. In Grefenstette, G., editor, *Cross-Language Information Retrieval*, chapter 2. Kluwer Academic Publishers, Boston.

Elkateb, F. and Fluhr, C. (1998). EMIR at the CLIR Track of TREC6. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Gachot, D. A., Lange, E., and Yang, J. (1998). The SYSTRAN NLP browser: An application of machine translation technology in multilingual information retrieval. In Grefenstette, G., editor, *Cross-Language Information Retrieval*, chapter 9. Kluwer Academic Publishers, Boston.

Gey, F. and Chen, A. (1998). Phrase Discovery for English and Cross-Language Retrieval at TREC6. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Grefenstette, G., Hull, D., Gaussier, E., and Schulze, B. (1998). Xerox TREC-6 Site Report: Cross-Language Text Retrieval. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Hull, D. and Grefenstette, G. (1996). Querying Across Languages: A Dictionary-based Approach to Multilingual Information Retreval. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland*, pages 49–57.

Landauer, T. and Littman, M. (1991). Fully Automatic Cross-Language Document Retrieval Using Latent Semantic Indexing. In *Proceedings of the 11th International Conference on Expert Systems and their Applications.*

Littman, M. L., Dumais, S., and Landauer, T. K. (1998). Automatic cross-language information retrieval using latent semantic indexing. In Grefenstette, G., editor, *Cross-Language Information Retrieval*, chapter 5. Kluwer Academic Publishers, Boston.

Mateev, B., Munteanu, E., Sheridan, P., Wechsler, M., and Schäuble, P. (1996). ETH TREC-6: Routing, Chinese, Cross-Language and Spoken Document Retrieval. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Nie, J. and Chevallet, J. (1998). Using Terms or Words for French Information Retrieval? In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Oard, D. and Hackett, P. (1998). Document Translation for Cross-Language Text Retrieval at the University of Maryland. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Peters, C. and Picchi, E. (1997). Using Linguistic Tools and Resources in Cross-Language Retrieval. In *Proceedings of the 3rd DELOS workshop; Cross-Language Information Retrieval, ERCIM Workshop Proceedings No. 97-W003, (ISBN: 2-912335-02-7), Zurich, Switzerland,*, pages 75–84.

Rehder, B., Littman, M., Dumais, S., and Landauer, T. (1998). Automatic 3-Language Cross-Language Information Retrieval with Latent Semantic Indexing. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Sheridan, P. and Ballerini, J. P. (1996). Experiments in Multilingual Information Retrieval using the SPIDER System. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland*, pages 58–65.

Sheridan, P., Braschler, M., and Schäuble, P. (1997). Cross-Language Information Retrieval in a Multilingual Legal Domain. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, Pisa, Italy*, pages 253–268.

Sheridan, P. and Schäuble, P. (1997). Cross-Language Multi-Media Information Retrieval. In *Proceedings of the 3rd DELOS workshop; Cross-Language Information Retrieval, ERCIM Workshop Proceedings No. 97-W003, (ISBN: 2-912335-02-7), Zurich, Switzerland,*, pages 65–74.

Smeaton, A., Kelledy, F., and Quinn, G. (1998). Ad hoc Retrieval Using Thresholds, WSTs for French Monolingual Retrieval, Document-at-a-Glance for High Precision and Triphone Windows for Spoken Documents. In *Proceedings of the Sixth Text Retrieval Conference (TREC6), National Institute of Standards and Technology (NIST), Gaithersburg, MD.*

Yamabana, K., Muraki, K., Doi, S., and Kamei, S. (1998). A language conversion front-end for cross-language information retrieval. In Grefenstette, G., editor, *Cross-Language Information Retrieval*, chapter 8. Kluwer Academic Publishers, Boston.

The TREC-6 Filtering Track: Description and Analysis

David A. Hull

Xerox Research Centre Europe
6 chemin de Maupertuis, 38240 Meylan France
hull@xrce.xerox.com

Abstract

This article details the experiments conducted in the TREC-6 filtering track. The filtering track is an extension of the routing track which adds time sequencing of the document stream and set-based evaluation strategies which simulate immediate distribution of the retrieved documents. It also introduces an adaptive filtering subtrack which is designed to simulate on-line or sequential filtering of documents. In addition to motivating the task and describing the practical details of participating in the track, this document includes a detailed graphical presentation of the experimental results and attempts to analyze and explain the observed patterns. The final section suggests some ways to extend the current research in future experiments.

1 Introduction

There is increasing evidence that text filtering will become a critical tool in searching and managing the flow of data in the information age. New companies are appearing daily which offer push services or intelligent agents centered around the core technology of content-based filtering. Since the beginning of TREC, the routing task has served as a forum for the development of these algorithms. However, as the uses of this technology have changed over the intervening years, the routing problem has gradually diverged from its most common applications. The filtering track was created to provide a more realistic simulation of the on-line time-critical applications of this technology.

We will start by describing the routing task as defined in TREC and use this definition as the starting point for the filtering task. Here is a simple definition of the routing task:

> Given a topic description and a large collection of documents, a sample of which have been evaluated as relevant or not relevant for that topic, construct a query profile and a routing function which will score and rank new documents according to their likelihood of relevance.

Filtering builds upon the routing task by introducing the concept of time. The routing task has traditionally been evaluated by average precision curves and other rank-based measures which use the complete ranked list of all incoming documents to determine performance. This list cannot be created until all documents have been scored and ranked, which means that system performance is not measured as a function of time. However, the underlying model upon which these tasks are based is one where a stream of transient documents are compared to a fixed set of query profiles. This means that the routing task simulates a non-interactive process where a user looks at documents only once at the end. A more realistic situation is one where users examine documents

periodically over time. The actual frequency of user interaction is unknown and task-dependent. Rather than attempt to simulate a particular task which might allow for batching and partial ranking of the document set, the filtering track operates on the opposite end of the spectrum, assuming that the user wants to be notified about each potentially interesting document immediately after it arrives. This leads to the following definition of the filtering task:

> Given a topic description, an incoming document stream, and possibly a small historical database of documents which have been evaluated as relevant or not relevant for that topic, construct a query profile and a filtering function which will make a binary decision to either accept or reject each new document as it arrives.

Since a decision to accept or reject a document must be independent of subsequently arriving documents, evaluation by rank-based measures is not appropriate. Filtering results will consist of unordered sets of documents which will be analyzed using set-based evaluation measures.

More background on the motivation for the filtering track and some scenarios for its use can be found in the TREC-5 filtering track description [2]. We recognize that there is still a considerable gap between the TREC filtering experiments and what goes on in operational systems. In particular, the task is fairly general, and their is no user interaction in the evaluation process. There is also no control of system performance in the area of efficiency (documents filtered per second) and scalability. This is because there are certain unavoidable trade-offs that must be made to make the filtering track compatible with the standard TREC experimental framework. However, there are many advantages to the TREC approach, the foremost of which is strong quantitative evaluation of system effectiveness.

2 TREC-6 Task Description

The TREC-6 filtering task has been substantially revised from the track definition and the experiments conducted in previous years, but the goals remain basically the same. The basic goals of this year's track are: to move towards a more realistic filtering task while retaining strong quantitative evaluation and to broaden the range of possible filtering experiments. The routing task has been criticized as unrealistic for a number of reasons. First, average precision and the other routing evaluation measures assume a post-hoc evaluation conducted over the entire test document collection. Second, while filtering has always been thought of as a temporal process, no time ordering or date information has been associated with the documents. Third, the training documents have come from a variety of different collections, most or all of which do not correspond to the source of the new test documents. Fourth, the size and coverage of the training set is much greater than one would find in most realistic applications. Fifth, the training documents have come from a variety of different systems, almost all of which are different from the filtering systems that will be tested. Of these five points, only the first has been addressed in previous iterations of the filtering track. The second and third point were addressed this year by the selection of the training and test documents. The fourth and fifth points are addressed in the adaptive filtering subtrack defined later in this section.

2.1 Topics and Documents

The corpus for the TREC-6 filtering experiments comes from the Foreign Broadcast Information Service (FBIS), which selects (and translates) text documents or transcripts from various non-American broadcast and print publications [4]. The 130,000 training documents date mostly from 1993 and early 1994 while the 120,000 test documents date come from late 1994 and 1995. All

documents have date stamps attached and have been ordered according to their date. The date stamps represent (more or less) the release date of the FBIS version rather than the release date of the original source document. Due to practical constraints (the first part of the data was released last year), the documents are not cleanly separated by date. Many test documents predate some of the training documents and some test documents predate most of the training documents. All systems were required to filter test documents in date order, allowing researchers to make use of the dimension of time. In addition, both the training and the test documents come from the same source, which is viewed as a more realistic simuulation of most filtering applications.

There were 47 topics for the TREC-6 filtering experiments, which are listed here:

```
  1   3   4   5   6  11  12  23  24  44  54  58  62  77  78  82  94  95
100 108 111 114 118 119 123 125 126 128 142 148 154 161 173 180 185 187
189 192 194 202 228 240 282 10001 10002 10003 10004
```

Of these topics, 38 were used in last year's routing/filtering experiments, and 9 of the topics are new this year (listed here):

```
62 128 148 180 282 10001 10002 10003 10004
```

The topics 10001-10004 were built specially for the routing/filtering task this year. The 9 new topics have incomplete relevance judgements on the FBIS training data. In particular, only the top 100 or so documents retrieved according to the NIST ZPRISE system were judged for relevance. Therefore, there are two different types of topics in terms of the relevance judgements. The 38 old topics have hundreds of relevance judgements on documents retrieved from many different sources while the 9 new topics have fewer than 100 relevance judgements on documents all retrieved by the same source. We will explore the impact of this dichotomy on the performance of different systems. The same topics were also used for the routing task. Note however, that routing systems are allowed to use relevance judgements for these topics from any of the TREC subcollections while filtering systems are limited to the FBIS training documents only. This represents an important divergence between the routing and filtering experiments which is happening for the first time this year.

2.2 Evaluation

Filtering systems are expected to accept or reject each document as it arrives and it is assumed that the user may well look at accepted documents immediately. Therefore, the output of the filtering system is treated as an unordered set of documents. This means that evaluation measures based on a ranked set of documents, such as precision-recall curves, are not appropriate. Instead, we apply two set-based evaluation metrics, utility and average set precision (ASP).

2.2.1 Utility and Average Set Precision

Utility assigns a value or cost to each document, based on whether it is retrieved or not retrieved and whether it is relevant or not relevant, as shown in the contingency table below:

	Relevant	Not Relevant
Retrieved	R+ / A	N+ / B
Not Retrieved	R- / C	N- / D

$$\text{Utility} = A*R+ + B*N+ + C*R- + D*N-$$

The variables R+/R-/N+/N- refer to the number of documents in each category. The utility parameters (A,B,C,D) determine the relative value of each possible category. A positive utility parameter can be thought of as the value of each document in that category, while a negative utility parameter is the cost of classifying a document in that category. Therefore, the larger the utility score, the better the filtering system is performing for a given query profile. For TREC-6, we test two different settings of the utility parameters:

```
F1 = 3*R+ - 2*N+        --> retrieve if P(rel) > .4
F2 = 3*R+ - N+ - R-     --> retrieve if P(rel) > .2
```

Filtering according to a utility function is equivalent to filtering by estimated probability of relevance. Therefore, the description above also shows the appropriate probability thresholds which correspond to the utility functions. Readers can find the general formula for converting a utility function into a probability threshold in Lewis [3], and a derivation of the formula can be found in any general book on decision theory.

Average set precision (ASP) is defined as the product of precision and recall, i.e. ASP = precision * recall. One can also think of ASP as a variant of average uninterpolated precision. Average uninterpolated precision computes the precision at the position of each relevant document in the ranked list, takes the sum, and divides by the total number of relevant documents. Average set precision is calculated on unordered document sets and uses the precision of the entire retrieved in place of the precision at each rank position. Otherwise, the calculation is the same, as this number is multiplied by the number of relevant documents retrieved and divided by the total number of relevant documents.

Utility is not an ideal measure for judging the performance of filtering systems for a number of reasons. First, utility scores will vary widely from topic to topic based on the number of relevant documents, and there is no valid way to normalize them, meaning that the scores cannot easily be averaged or compared across topics. Second, utility treats all relevant documents as equally important, no matter how many have already been retrieved or how many exist for a given topic. There is no sense of diminishing returns, which seems counter-intuitive in many situations. Third, a user would probably have a different utility function for each topic, based on the topic's difficulty and the number of relevant documents. For practical and administrative reasons, it is too difficult to define a separate utility function for each topic.

However, ASP also has its problems. In particular, when no relevant documents are returned (and relevant documents exist), a system receives a score of zero, irrespective of the size of the retrieved set. This means that retrieving no documents is equivalent to retrieving an arbitrary number of non-relevant documents. Clearly, the former result is much preferred over the latter, and a major part of the challenge in text filtering is knowing when not to retrieve any documents. It should be noted that van Rijsbergen's E-measure also suffers from this problem. So far, no one has developed a good general measure of set-based retrieval effectiveness that does not suffer from at least some of these drawbacks. A better approach for the future may be to select a particular task and a particular user model, then define a model- and task-specific measure.

2.2.2 Pooling vs. Sampling

In general, the size of the retrieved set is unbounded, making accurate evaluation of performance difficult for some topics, regardless of measure. The traditional pooling approach to document assessment has been augmented with a random sampling strategy. In pooling, the top N retrieved documents from each run and each topic are merged to create a single document pool which is assessed. All documents which do not appear in that pool are assumed to be not relevant. This

strategy is reasonably fair for all participants, but results in performance estimates that are biased downwards. For filtering, the retrieved set is unranked, so one cannot simply select the top N documents. The approach for filtering is therefore to select a random sample of size N from the retrieved set for each system. If the retrieved set is smaller than N, all documents are selected.

Pooling is less than ideal for filtering topics where the retrieved sets are much larger than N. First of all, the filtering pool is of lower quality because the documents are randomly sampled from a large retrieved set rather than obtained by selecting the top ranked documents. Fortunately, this effect is mitigated somewhat in the TREC-6 experiments because the routing runs, which are based on ranked retrieval, also contribute to the pool. Second, the topics with large retrieved sets are the ones which will tend to have the most relevant documents, and thus will suffer from the most bias due to incomplete assessment. Fortunately, we know from sample theory that the proportion of relevant documents in a simple random sample is an unbiased estimate of the proportion of relevant documents in the population. For utility function F1, we can convert an estimate of the proportion of relevant documents directly into an estimate of utility via the following formula ([2], p.81, eq 1):

$$\text{F1} = ((A - B) * (r/n) + B) * N$$

where n is the size of the sample, r is the number of relevant documents in the sample and the other terms are the same as defined above. Another nice property of the sampling approach to evaluation is that we can also calculate the standard error of the utility estimate, which is given by ([2], p.87, eq 30):

$$\text{SE(F1)} = ((A - B)^2 N \frac{(N - n)r(n - r)}{n^2(n - 1)})^{1/2}$$

Unfortunately, sampling cannot provide us with an estimate of recall, which is necessary for F2 Utility and ASP, so these measures can only be estimated via pooling. For more details on the mathematics of sampling, see Lewis [2]. Sampling provides unbiased estimates of utility, but there is a price. Since a lot of information is thrown away (all the relevance judgements for documents sampled by other systems but not the one being evaluated), the sampled estimates tend to have much more variation.

2.3 Basic Task Summary

Each participating group could submit up to two filtering runs for each of the three evaluation measures: F1 Utility, F2 Utility, and ASP. Participating groups were required to submit at least one F1 Utility run and at least one F2 Utility run. Runs were classified into one of three categories:

(A) Automatic - Any run which uses fully automatic methods for profile construction and updating. This can include automatic learning from test documents as they are filtered.

(B) Manual - Any run which uses manual techniques for profile construction, up to and including making additional relevance judgments on training documents. No manual intervention based on information from the test documents is allowed, although automatic learning is still permitted.

(C) Manual Feedback - Any run which uses manual techniques for updating profiles based on previously viewed test documents. The run may or may not also use manual techniques for profile construction.

In practice, only one manual run was submitted and no one submitted any runs based on manual feedback. Previous TREC routing and filtering experiments have found no consistent advantage

to using manual techniques for profile construction, so all submitted runs were treated as a single category. For most topics, there are enough evaluated documents already that the cost (in terms of human effort) of additional manual assessment would not be justified by the returns. In addition to the three evaluation measures, there were two optional subtasks that groups could participate in, as will be described in the next two subsections.

Due to assessment constraints, NIST promised to evaluate a maximum of 100 documents per topic per participating group. These documents were obtained as follows. Only the mandatory runs contributed to the assessment pool. If the F1 Utility set for topic T is greater than 100 documents, 100 documents are sampled for assessment. Otherwise, all documents are assessed. Furthermore, a sufficient number of document from the F2 Utility set are sampled to bring the total set size up to 100 documents. If the union of the F1 Utility and F2 Utility sets is less than 100 documents and the group submitted a ranked retrieval run (see the next subsection), then the sampling program went down the list in rank order adding new documents until a sample of size 100 was extracted. This final step was added to improve the quality of the document pool as much as possible given the assessment constraints.

2.3.1 Comparison to Ranked Retrieval

One can view the routing task as a subset of the filtering task, where routing consists of defining a document scoring function which tends to rank relevant documents above non-relevant ones, while filtering consists of building a thresholding function on top of the scoring function to optimize some set-based evaluation criterion. In the past, most systems participating in filtering have also participated in routing and used the same scoring function for both tasks. For systems which follow this model, we can try to some extent to separate the performance of the thresholding function from the performance of the scoring function. Participation in this subtask is optional, since there is no a priori reason why a system must use a scoring algorithm in their filtering system. One could use a binary rule-based algorithm instead. In addition, more complex systems may change the scoring function as they view test documents, meaning that scores generated at the beginning of the document stream may not be comparable with scores generated later on in the document stream.

For TREC-6, we try to learn about this issue by applying two different simple tests. For each test, we take the ranked list of documents returned by the scoring function and calculate the score obtained by selecting the optimal threshold. This is computed by exhaustive search over the ranked list of relevant documents. In the first test, we ask the simple question, did the system overestimate or underestimate the threshold? By accumulating this information over the full topic set, we gain a general sense about whether the system is biased towards retrieving too many or too few documents. In the second test, we compare performance based on the optimal thresholds to the observed performance to determine whether this changes the relative ranking of participating systems.

2.3.2 Adaptive Filtering

Previous versions of the TREC filtering task have been viewed as unrealistic because they provide too much training data, and because this training data comes from the previous search results of many different systems which use many different search algorithms. In practice, most systems can only expect relevance judgements from documents that they have been responsible for retrieving. The adaptive filtering subtrack is designed to model this situation. In this task, each system starts only with the topic description and no evaluated documents. Documents arrive sequentially and the

system can update the query profile in response to previously viewed documents. In addition, each document retrieved will be immediately evaluated for relevance, and that information will be passed on to the system. It is not possible within the TREC framework to actually provide new relevance judgements to filtering systems as they see documents. Instead, the interactive component must be simulated using the training data and the previously-released relevance judgements. In this model, the relevance judgement is available but hidden from the system until it decides whether or not it will retrieve a document. Relevance judgements from unretrieved documents are never revealed to the system. Note that test documents are evaluated in the same fashion, but relevance judgements are not immediately available (unless the system has a user providing manual feedback).

This means that systems may wish to evaluate a document not only according to its likelihood of relevance, but also according to its value as a training observation to improve future filtering performance. This makes filtering a more interesting and more complex task. Since many of this year's topics have only been partially evaluated over the training document collection, they are not suitable for adaptive filtering. Therefore, this subtask will use only the 38 TREC-5 topics which have been fully evaluated. Systems may choose whether they wish to use the rest of the TREC document collection (excluding the FBIS training document set) to generate collection frequency statistics (such as IDF) or auxilliary data structures (such as automatically-generated thesauri) or begin with no prior information. Systems may also decide whether they wish to treat unevaluated training documents as not relevant or whether they assume that the user simply declined to make a judgement in this situation. Performance will be judged on both the training and the test document set, and the results will not be compared directly to the other filtering tasks. This is because adaptive filtering runs will operate at a substantial disadvantage, since they only have access to relevance judgements from documents which they retrieve. Performance on the training set must also count to prevent groups from retrieving far too many documents on the training set to increase the pool of relevance judgements available to the system and thus improve performance on the test documents. Since this task is in an experimental stage, participants are also free to choose from among the evaluation measures.

3 TREC-6 results

The TREC-6 filtering track had 10 participating groups who submitted a total of 59 runs, divided as follows:

	# groups	# runs
Total	10	59
-----	--	--
F1	10	17
F2	10	17
ASP	7	15
ranked	7	11
adaptive	1	2

Note that the sum of the runs column adds up to more than the total number of runs because some groups submitted the same run for more than one evaluation measure. The participating groups [abbreviations] (run identifiers) were AT&T Labs Research [AT&T] (att97f), Australian National University [ANU] (anu6flt), City University London [City] (city6f), CLARITECH Corporation [CLARITECH] (CLRoute/CLComm), Daimler Benz AG Research Center Ulm [Daimler Benz] (dbulm1), Queens College CUNY [CUNY] (pircsF), Siemens AG [Siemens] (teklis6), University of

California Berkeley [Berkeley] (BKYT6f), University of Massachusetts Amherst [UMass] (INQ4), University of North Carolina [UNC] (isf).

3.1 Summary of approaches

In this section, we briefly describe the techniques used by each of the groups for the filtering track[1]. For more information, please consult the individual participants' papers included in this volume [5]. Almost all participants treat filtering as a special case of routing, using a ranked retrieval system, followed by thresholding based on the document score. The vast majority of systems use the same basic technique for finding the threshold: observe the score on the training set where the evaluation measure is maximized and use that score as the threshold for the test set. The exceptions will be noted below. Several system also use logistic regression to convert the scores to probability estimates and filter on the probability estimates directly. The major differences between systems come from their scoring algorithms.

AT&T builds a feature set of terms, phrases (adjacent pairs), and non-adjacent pairs based on the term weights from Rocchio expansion. It then optimizes the feature weights for average uninterpolated precision using a technique called Dynamic Feedback Optimization (DFO). Their experimental run (att97fe) takes the lowest scoring half of the relevant documents and the highest scoring non-relevant documents based on the initial query and constructs a second query via the same methods. The final routing profile is a weighted combination of these two queries.

ANU selects terms and phrases (adjacent pairs) from the topics and from the training documents using independent algorithms. The topic selection algorithm assigns higher weight to title words, words with high frequency, and words which are all in capitals. The document selection algorithm scores terms based on the difference between their probability of occurrence in the relevant documents and their probability of occurrence in the non-relevant documents. The number of features and the relative weight of the feature set returned by each algorithm are then passed on to a Generalized Reduced Gradient (GRG2) nonlinear optimization program, which optimizes these parameters with respect to the utility measures.

City orders terms and adjacent pairs based on their weight according to a probabilistic model, then applies forward stepwise feature selection using a fixed increase in average uninterpolated precision as the selection criterion. Term weights are then optimized using two methods: deterministic adhoc weight adjustment and a simulated annealing procedure. In each case, the training set is partitioned into two halves; the term selection is based on one half while the weight optimization is based on the other half. The partitions are then inverted and the procedure is repeated. Additional repetitions are generated by splitting the training set randomly into different partitions. All the resulting term sets are merged to build the final query profile. The difference between their two runs is that city6f1 optimizes weights on half the database and selects thresholds on the other half while city6f2 optimizes weights on both halves and selects thresholds on the entire database.

CLARITECH uses a probabilistic model in the first pass for term selection followed by a second pass of the Rocchio algorithm for term selection and term weighting. Thresholds are chosen via logistic regression for F1 Utility and using the optimal score on the training documents for the other measures. CLARITECH tested both threshold selection methods on the training set and then picked the one that worked best for each measure. CLROUTE is based on this method while CLCOMM divided the training data into two parts and applied the terms selection algorithms to each part separately. Only the terms that appeared in both sections were retained for the final profile.

[1]This section is based on the material provided by the groups in their draft papers.

Daimler Benz generates 5 different feature sets: 2 based on n-grams (3- and 4-grams), 3 based on substrings extracted from terms (a stemming variant) and selected for using the following measures: chi-square, TF-IDF, and correlation (with relevant documents?). Each feature set is transformed and reduced in dimension (to 600 for filtering) by taking an eigen decomposition of the feature covariance matrix (Principal Component Analysis - equivalent to Latent Semantic Indexing). Filtering is treated as a 47-class categorization problem (each topic is one category) and a single classification rule is built for all categories simultaneously using linear regression. The final profile is created by merging the decision vectors constructed for each feature set.

CUNY combines two runs, one based on terms (and term pairs?) extracted from the topics, and one based on terms (and term pairs?) extracted from the training documents, in their submission. The only difference between the two submissions is that one of them lowers the threshold by 10% to reflect the fact that subsequent relevant documents are likely to be less similar than training documents which are used both for profile creation and for threshold selection.

Siemens uses a true filtering system which computes document scores based on term correlations and a probabilsitic model. The feature set is single terms only and the topic statement is not used.

Berkeley first performs term selection using a chi-square measure (positive association with relevance only) with a significance cut-off of 0.001. These terms are put into an expanded query which is then scored against the training documents. In addition, terms are ranked according to their log-odds of relevance and the top 5 terms are selected. The RSV scores from the first step and the log term frequencies of the 5 terms extracted in the second step are passed as parameters for logistic regression, which produces the final probabilistic scoring function.

UMass builds an initial query based on the topic statement and extracts the top scoring 200 word passage from each evaluated training document. The feature set consists of the top 20 each of terms, phrases, and pairs in a 20-word window, according to the difference between probability of occurrence in relevant documents and probability of occurrence in non-relevant documents. These features are weighted and the weights are optimized for average uninterpolated precision using Dynamic Feedback Optimization. For adaptive filtering, UMass uses a true filtering system which builds a profile based on the initial topic statement and updates it with an incremental Rocchio algorithm. Thesholds are set at the mid-point between the average score on the relevant documents and the average score on the non-relevant documents.

UNC creates an "information space" for each topic by applying Principal Components Analysis (an eigen decomposition) to a co-occurrence matrix constructed from all query terms. Training documents with at least 25% of the query terms are located in this space. The threshold for F1 Utility was chosen to be the minimum distance between any evaluated document and the topic statement in the information space. Any document with at least 25% of the query terms was retrieved for the runs based on F2 Utility.

3.2 Comparative Evaluation

When evaluation is based on utility measures, it is difficult to compare performance across topics. Simple averaging of the utility measure gives each retrieved document equal weight, which means that the average scores will be dominated by the topics with large retrieved sets (as in micro-averaging). Therefore, comparative evaluation will be based on an average rank measure which treats all topics equally. This measure is computed in two steps: (1) for each topic, rank the systems in order of their performance, (2) average the ranks by system over all topics. This means that the larger the average rank score, the better the system is performing with respect to its competitors. Table 1 presents a pseudo-example of the ranking process.

The average rank measure has its advantages and disadvantages. On the positive side, all

Utility	R1	R2	R3	R4
T1	0	0	-18	4
T2	150	276	160	75
T3	-6	-44	-43	-11

Rank	R1	R2	R3	R4
T1	2.5	2.5	1	4
T2	2	4	3	1
T3	4	1	2	3

Table 1: Pseudo-example of ranking runs R1-R4 for topics T1-T3.

topics are equally important in determining a system's performance, meaning that the scores are insensitive to outliers or topics which may have high variation due to random factors. Average rank scores generated by the same set of systems are directly comparable, even if they are based on different evaluation measures or different retrieval tasks. This allows one to do a global comparative evaluation in situations where it would otherwise be difficult (as is the case with utility). On the negative side, all topics are equally important, meaning that topics with large variation which reflect real differences between systems do not receive higher weight. There is no absolute standard of performance, the scores are only meaningful relative to the performance of different systems. The results depend on the systems being compared, so adding or removing a system will change the results of other systems.

Average Set Precision (ASP) is comparable across topics, and will be evaluated both using the traditional average score and the average rank measure. As mentioned previously, ASP does not distinguish between systems which retrieve no relevant documents. However, using ranks allows us to introduce a tie-breaking procedure. Systems which have ASP = 0.0 for a given topic will be ranked in inverse order of the number of documents which they retrieve.

We can use non-parametric statistical tests to determine the significance of average rank differences between systems. In order to keep the results simple and readable given the large number of experiments and competing systems, we will limit ourselves to pairwise comparisons with respect to the best performing system. For each experiment, we apply two different tests: a conservative test (a non-parametric variant of the Newman-Keuls multiple comparisons test) and a more powerful test (a non-parametric variant of the Least Significant Difference (LSD) test). For more information on the statistical tests and their performance characteristics in TREC-style IR experiments, please consult our NIST technical report [1]. The alpha level is set at 0.05, which corresponds to the error rate for each pairwise comparison in the LSD. The true pairwise error rate for the Newman-Keuls is an order of magnitude smaller.

3.3 Evaluation results

Figures 1-8 present the evaluation results for the TREC-6 filtering experiments. Each figure contains two or more experimental runs drawn on the same page. Dashed lines are drawn to link the runs from the same system. In most cases (but not always), these will correspond to the same underlying retrieval algorithm with a different threshold or thresholding strategy. The vertical arrows link all the systems which are not distinguishable from the top performing system according to the statistical significance tests described in the previous section. The longer arrow always corresponds to the more conservative test.

We should mention that two groups had problems with their submitted runs. CUNY (pircs) optimized for the wrong utility thresholds, so their F1 and F2 Utility runs are not accurate. Daimler Benz (dbulm) did not understand how sampling would be used for evaluation and artificially truncated all of their runs to 100 documents. This will have a much higher impact on F2 Utility and ASP than on F1 Utility. Both groups were given the opportunity to submit revised results, and a revised (unofficial) comparison is presented in Figure 8.

Figure 1 compares system performance for F1 and F2 Utility. The system rankings remain more or less the same, with a few small shifts. The results indicate that there is not a lot of interaction between system performance and filtering threshold. Figure 2 compares system performance over the two utility measures and average set precision. Other than the two systems with errors in their runs (pircs and dbulm), there are not a lot of differences. We do notice that UMass (INQ) and Siemens (teklis) do better on the utility measures than ASP. These systems retrieved a uniformly small number of documents for all topics. Small retrieved sets are often good for the utility measures but rarely good for ASP.

Figure 3 compares evaluation based on average score to evaluation based on average rank for ASP. The average rank scores are rescaled to match the end points of the average precision scores. We find that both measures yield basically the same results. Figure 4 compares the pooling versus the sampling strategy for evaluation. Note that the results are presented only for the 12 topics where significant sampling took place. For all other topics, the retrieved sets are almost always less than 100 documents, so the pooled and the sampled results are virtually identical. Since the score differences over these 12 topics are also minimal, it is fair to conclude that pooling introduces no bias in favor of any individual system. In most cases, the pooled scores fall within the 95% confidence intervals for the sampled results. There are three topics (142, 189, and 202) for which the pooled scores fall outside the sampling confidence bounds with some regularity. In all these cases, the pooled estimate is significantly lower than the sampled estimate, indicating that there are probably still a fair number of unevaluated relevant documents for these topics.

Along with their filtering runs, groups were also allowed to submit a ranked list of documents which would correpond to the output of their filtering system without thresholding. By comparing this ranked list to the evaluation functions, we can determine the optimal threshold in a retrospective fashion. Figure 5 compares observed performance to performance based on choosing the optimal threshold for F1 Utility. Once again, we note very little difference between the two graphs. This should not be surprising since almost all participants used more or less the same strategy for threshold selection. CLARITECH did a direct comparison of the two most common strategies for threshold selection: choosing the retrieval score which optimizes performance on the training set and logistic regression. For CLRoute, they found that logistic regression worked better for F1 Utility while optimal score worked better for F2 Utility. For CLComm, they found that the two worked about the same. A more detailed analysis of threshold bias is presented in the next section.

The TREC-6 filtering topics can be divided into two categories: the 38 old topics which were heavily evaluated over the training FBIS data at TREC-5 and the 9 new topics, where evaluation over the training data is limited to roughly 100 documents retrieved by the PRISE system. Figure 6 breaks down the average rank scores into the old and new topic sets for F1 Utility. Performance remains roughly constant for most systems (ignoring the pircs runs which have errors). However, the cityf2 run jumps upward by more than one rank position. While the topic sample is too small to say that the difference is statistically significant, one can speculate that the approach adopted by City might have some advantage for small training sets. While cityf1 does not show the same jump, this result can be easily explained. cityf1 uses only half the training data for optimization, reserving the other half for threshold selection. With smaller training sets, one might expect that ignoring half the data would offset any other advantages inherent to the technique.

It is interesting to examine whether there is any interaction between system performance and the inherent difficulty of the topics. For this experiment, we defined difficulty according to the median F1 utility score for the topic. All topics with a median utility greater than zero were classified as easy and all topics with a median utility less than zero were classified as hard. This resulted in 21 easy and 17 hard topics (the remainder had median utility zero). Figure 7 breaks down performance as a function of easy and hard topics. We note immediately that the systems

which retrieved few documents for all topics do much better on the hard topics than the easy topics. This reflects the fact that for the hard topics, it is generally correct to retrieve very few documents. The opposite pattern emerges for three of the four top-ranked systems. These systems are getting almost all of their gain on the easy topics and are performing significantly worse than most of the remaining systems on the hard topics. This means that these systems (cityf1, cityf2, att97fc) need to work on being more robust for hard topics. The only exception is att97fe, which performs well for all topic categories. We should note that only 2 of the 9 new topics are hard, meaning that the size of the training set is not strongly linked to topic difficulty.

Figure 8 shows the revised comparisons after Daimler Benz (dbulm) and CUNY (pircs) submitted corrected results. It is reassuring to note that the relative positions of all the other systems remain roughly the same.

3.4 Absolute Performance and Threshold Bias

The graphs presented in the previous section are missing a key component. They tell us about the relative performance of systems but say nothing about performance according to an absolute standard. It is hard to decide on a standard which is appropriate for all topics, so we fall back on a very simple approach for the F1 utility measure. A system which retrieves no documents at all receives an F1 utility score of zero. Therefore, any system which has a positive utility score for a given topic is providing some added value. This may seem like a fairly minimal standard, but it is sobering to see how hard it is in practice to measure up to this standard. The median TREC-6 filtering system has positive utility for 20 topics, zero utility for 11 topics, and negative utility for 16 topics. The best TREC-6 filtering system has positive utility for 33 topics. This means that the typical TREC-6 filtering system can barely justify its own existence. The same question is more complex for F2 utility, since systems are penalized for unretrieved relevant documents, so retrieving no documents leads to a negative score. However, we can compute how many systems have a utility score which is greater than what they would receive by retrieving no documents. The median system does better than the empty document set on 42 topics and worse on 5 topics, while the best system is better on 46 topics and worse on 1 topic. With respect to ASP, it is always better to retrieve at least some documents.

When we look at the actual utility scores, we are left with the impression that filtering systems in general are performing quite poorly. For example, the typical system has a positive F1 utility for slightly over half the topics and positive F2 utility for only 10 of 47 topics. There are two possible explanations for these results: (1) threshold selection really is a very hard problem, and/or (2) the filtering thresholds (and/or the measure itself) are simply an unrealistic standard of performance. We can get at this question by looking at the utility scores obtained by using the optimal thresholds. The optimal performance of the typical system results in positive F1 utility for about 35 topics and positive F2 utility for about 16 topics. The optimal performance of the best systems results in positive F1 utility for about 40 topics and positive F2 utility for about 30 topics. From these results, it seems that both factors are important. A system must be performing very well to return positive utility for the majority of topics, particularly for the F2 measure. Penalizing for unretrieved relevant documents is setting a very high standard. Perhaps it might be better to penalize only for retrieved non-relevant documents. On the other hand, there is clearly still a lot of room for improvement in threshold selection algorithms.

We can also use the optimal thresholds derived from ranked retrieval to determine if there is bias in the threshold selection algorithms. For each run, we simply ask whether the observed threshold is lower, higher, or the same as the optimal threshold and count up the number of topics which fall into each category. Table 2 presents the results for the median TREC-6 filtering system. We

Median	too few	exact	too many
F1	21	5	20
F2	28	2	17
ASP	37	1	9

Table 2: Number of topics where systems retrieved too few, the right number of, or too many documents

note immediately that there is very little bias for F1 Utility, but systems have a strong tendency to retrieve too few documents for F2 Utility and ASP. We can only speculate as to why this might be the case. The training set is not comprehensively evaluated so there will always be missing relevant documents. If systems underestimate the density of relevant documents on the training set, they are likely to do the same on the test set, which may lead them to select thresholds which are too restrictive. Most systems use the training documents both for building the filtering profile and for threshold selection. Therefore, the scores of relevant training documents will be biased upwards and this bias may be passed on to the threshold selected by those systems. For example, CLComm attempts to reduce this bias by splitting the training data into two parts and merging profiles built on both parts independently, while CLRoute does not. The results confirm that CLComm has less threshold bias than CLRoute for both F2 Utility and ASP.

4 General Commentary

Almost all groups attack the filtering problem by building a good ranking algorithm and then applying thresholding. This is partly because people are using the same systems for both routing and filtering. However, the best scoring systems optimize the ranking for some completely independent measure like average uninterpolated precision and still have a great deal of success on the filtering track. There does not seem to be any need to optimize the ranking specifically for the filtering evaluation measures. The key question is then how to set the filtering threshold.

The systems concentrate on two different threshold selection strategies: finding the score which optimizes performance on the training set or normalizing the scores to create probabilities using logistic regression and filtering on probability thresholds. We'll briefly describe the advantages and disadvantages of each approach. One can think of each method as looking at a plot of utility as a function of retrieval score on the training data and picking the maximum. The difference is that logistic regression smooths the curve before finding the maximum, which can be a significant advantage. Empirical estimation can suffer when the curve is extremely bumpy since the observed maximum may be only an artifact caused by an unusual clumping of relevant documents. Also, consider the case where the training data is poor (low density of relevant documents). The desired threshold score may be greater than the observed score of any of the relevant documents. Logistic regression allows the system to extrapolate a threshold which is greater than the score of the top-ranked document. Empirical methods provide no direct way to estimate a threshold outside the range of the observed data. On the other hand, logistic regression is fitting a parametric curve to data which may not have the logit distribution. This can result in biased estimates in some cases. CLARITECH demonstrates that logistic regression doesn't always work better than empirical methods.

There are alternative methods for smoothing the utility curve which have not yet been explored. Instead of logistic regression, one could adopt a very simple local smoothing algorithm. For example, one could calculate and average utility over a sliding window of n rank positions (or a range of

scores [x-a,x+a]) and choose the threshold as the midpoint of the window at the position where the average utility is maximized.

Many of the groups recognized the danger of using the same training set for both profile construction and threshold selection. Relevant documents used to construct the profile will have a much greater similarity to the profile itself than relevant documents which appear in the test data. As discussed in the previous section, this may lead to biased thresholds, a problem which is common to all methods of threshold selection. Both City and CLARITECH experiment with partitioning the training data into independent segments (City uses many different partitions), but they find that the runs based on the partitioned data tend to be less successful than runs based on the full data. This is probably because the runs based on partitioned data have fewer relevant documents to work with for profile construction and optimization. Perhaps there is a way around this trade-off. If the partitioned data set can be used to estimate the amount of bias directly, one could then retrain the system on the full data set and apply the derived bias correction factor to obtain the final thresholds.

There are still many more factors to explore in future filtering experiments. For example, none of the groups made real use of the time ordering of the training data. It is possible that the distribution of a term over time may be related to its value as a feature in the filtering profile. Terms occuring frequently over a short period of time may reflect a single event and thus not have enduring predictive value. One can also model the density of relevant documents as a function of time, which is a crucial factor in threshold selection. Even if the scoring algorithm is optimal and the distribution of relevant and non-relevant documents is the same over both the training and the test data, the ideal threshold can change substantially depending upon the relative density of relevant and non-relevant documents. Experiments in adaptive filtering are only beginning. Only one group (UMass) tried adaptive filtering for TREC-6. Readers are referred to their paper for more details about the methods and the results.

The filtering track will continue at TREC-7. In fact, it will be expanding, as the routing task will be folded into the filtering track next year. There will be an increased emphasis on adaptive filtering, as this is thought by most to be a more realistic problem, and it provides the new challenge of sequential or on-line learning. However, routing and batch-style filtering will continue to exist to support research in traditional text categorization and machine learning algorithms. We encourage everyone with interest in the subject to participate in the filtering track next year.

Acknowledgements Many people have contributed to the development of the TREC filtering track, in particular David Lewis, Karen Sparck Jones, Chris Buckley, Paul Kantor, Ellen Voorhees, the TREC program committee, and the team at NIST. I am merely building upon their work.

References

[1] David A. Hull and Paul B. Kantor. Advanced Approaches to the Statistical Analysis of TREC Information Retrieval Experiments. Contact the first author for a copy: hull@xrce.xerox.com, 1997.

[2] David Lewis. The TREC-5 Filtering Track. In *The 5th Text Retrieval Conference (TREC-5), NIST SP 500-238*, pages 75–96, 1997.

[3] David D. Lewis. Evaluating and Optimizing Autonomous Text Classification Systems. In *Proc. of the 18th ACM/SIGIR Conference*, pages 246–254, 1995.

[4] Ellen M. Voorhees and Donna Harman. Overview of the 5th Text Retrieval Conference (TREC-5). In *The 5th Text Retrieval Conference (TREC-5), NIST SP 500-238*, pages 1–28, 1997.

[5] Ellen M. Voorhees and Donna Harman, editors. *The 6th Text Retrieval Conference (TREC-6)*, 1998. To appear.

Figure 1 - F1 vs. F2 Utility

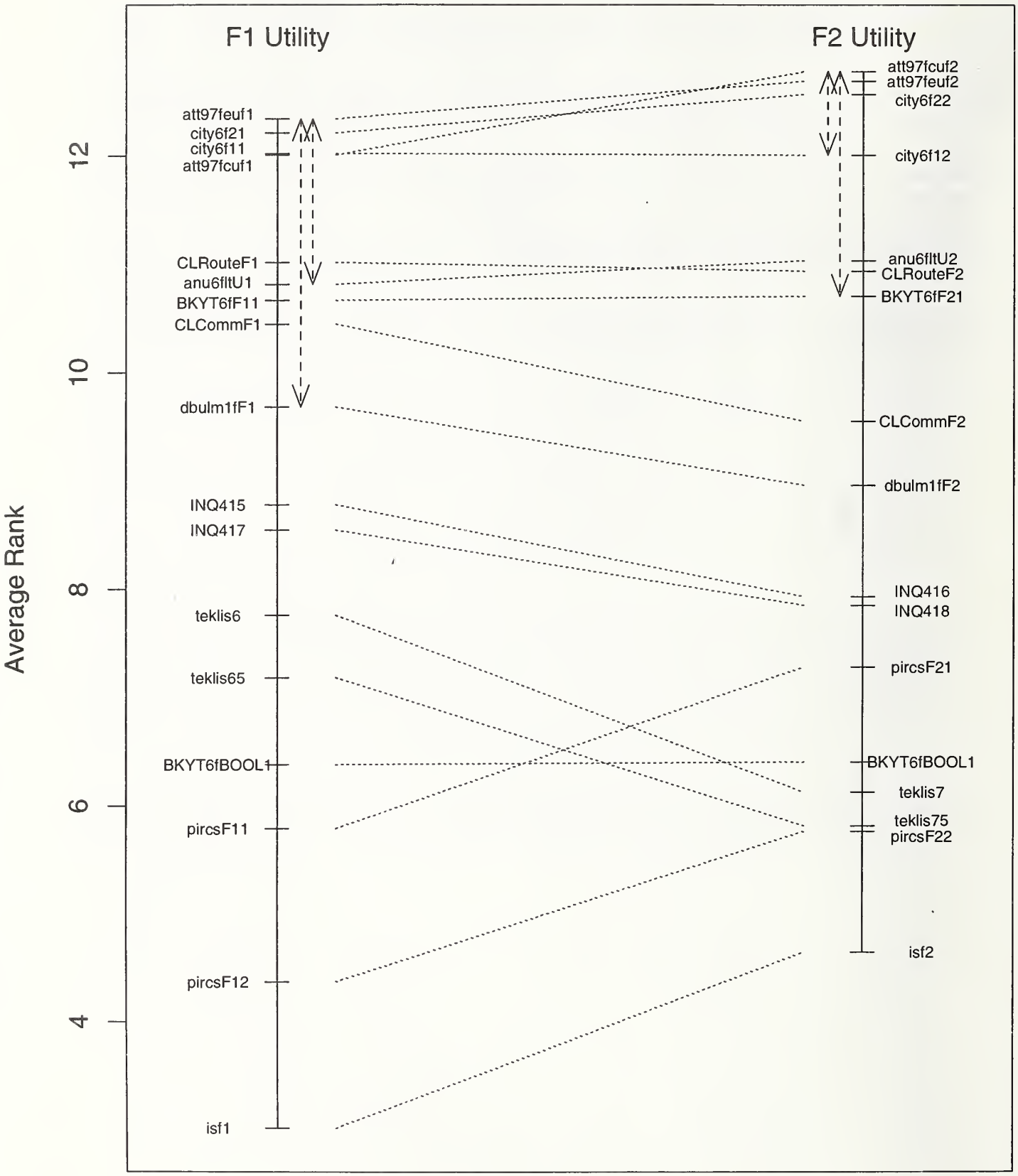


Figure 2 - Utility vs. Average Set Precision

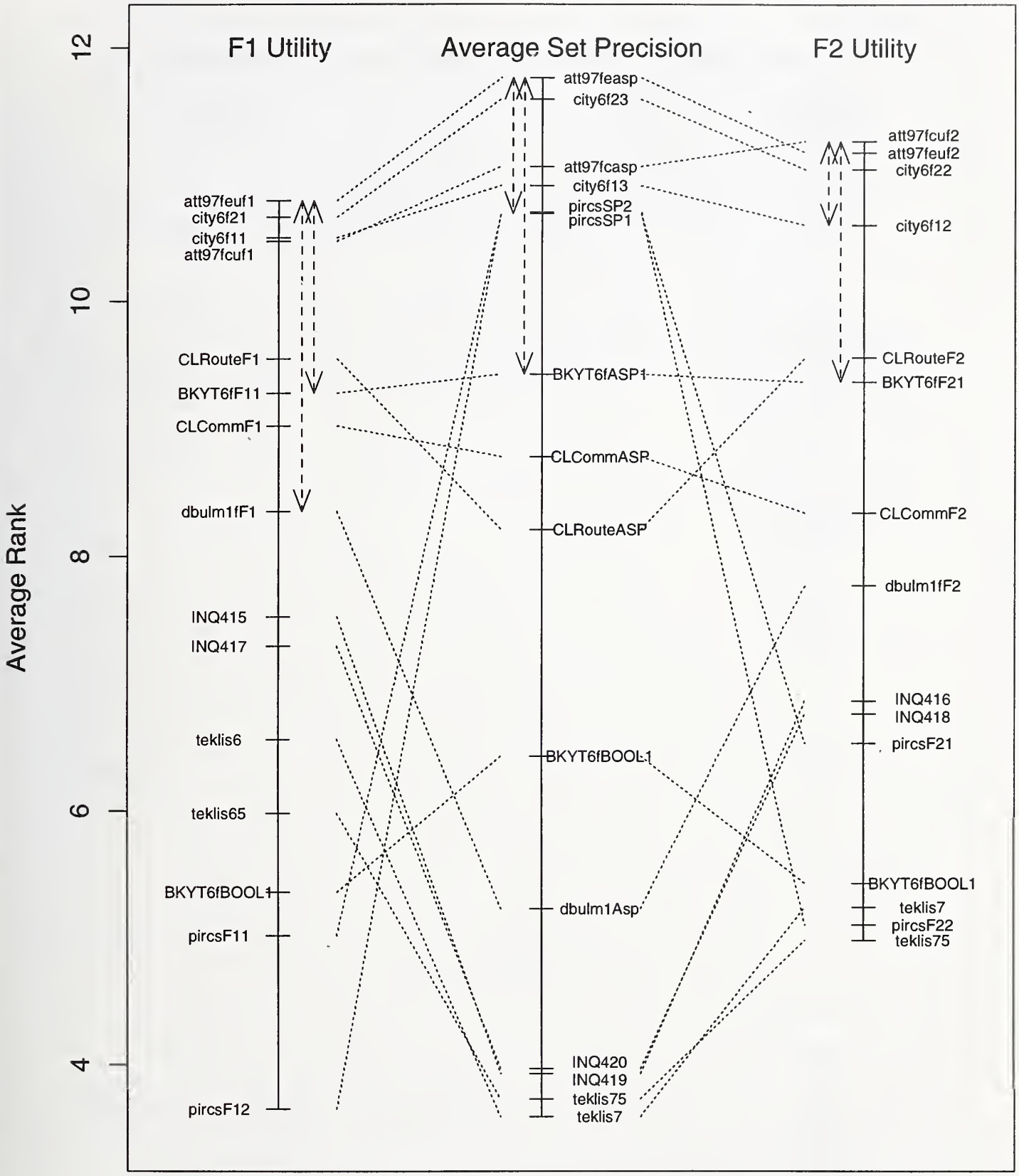


Figure 3 - Averaged vs. Ranked ASP

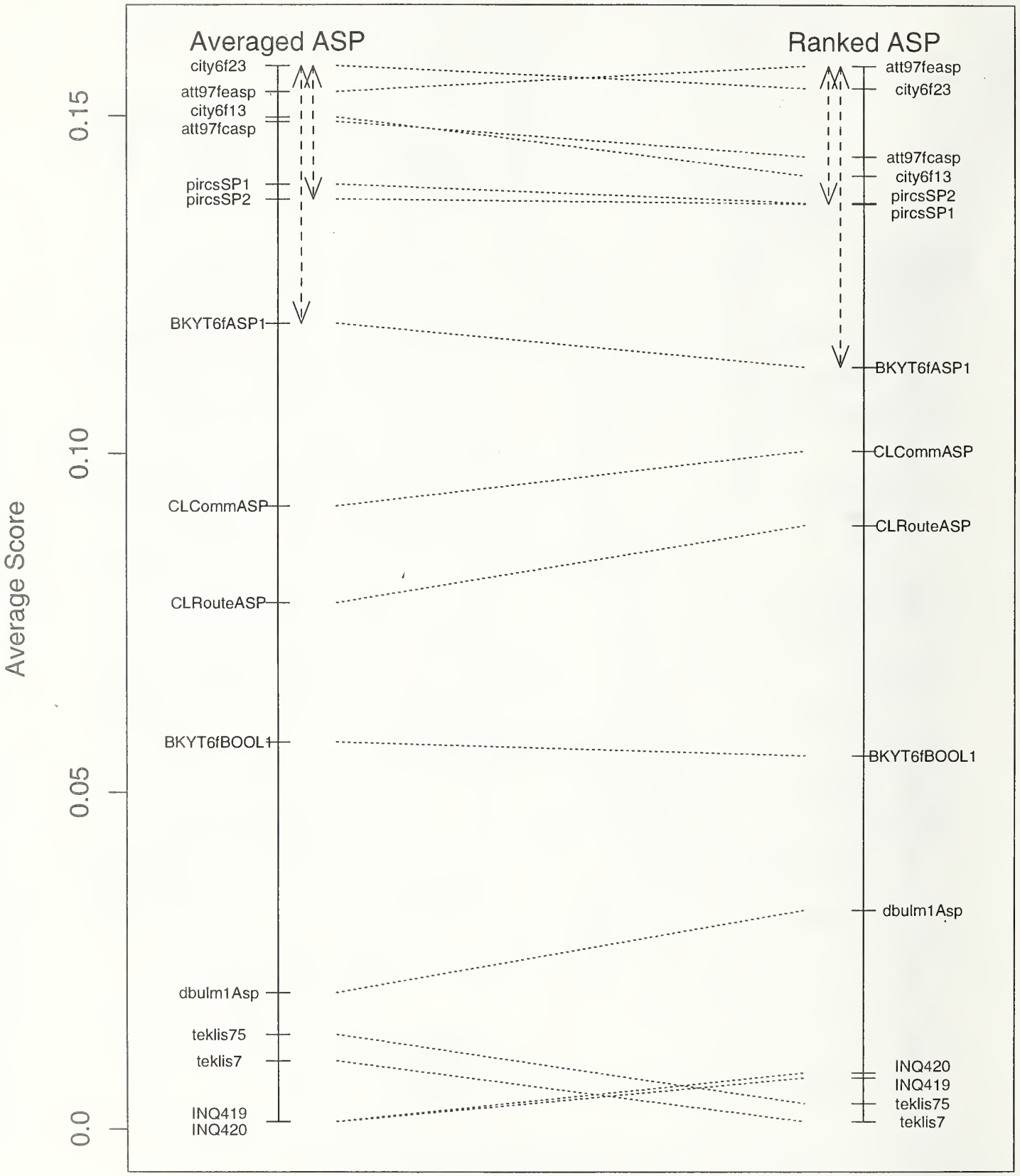


Figure 4 - Pooled vs. Sampled F1 Utility

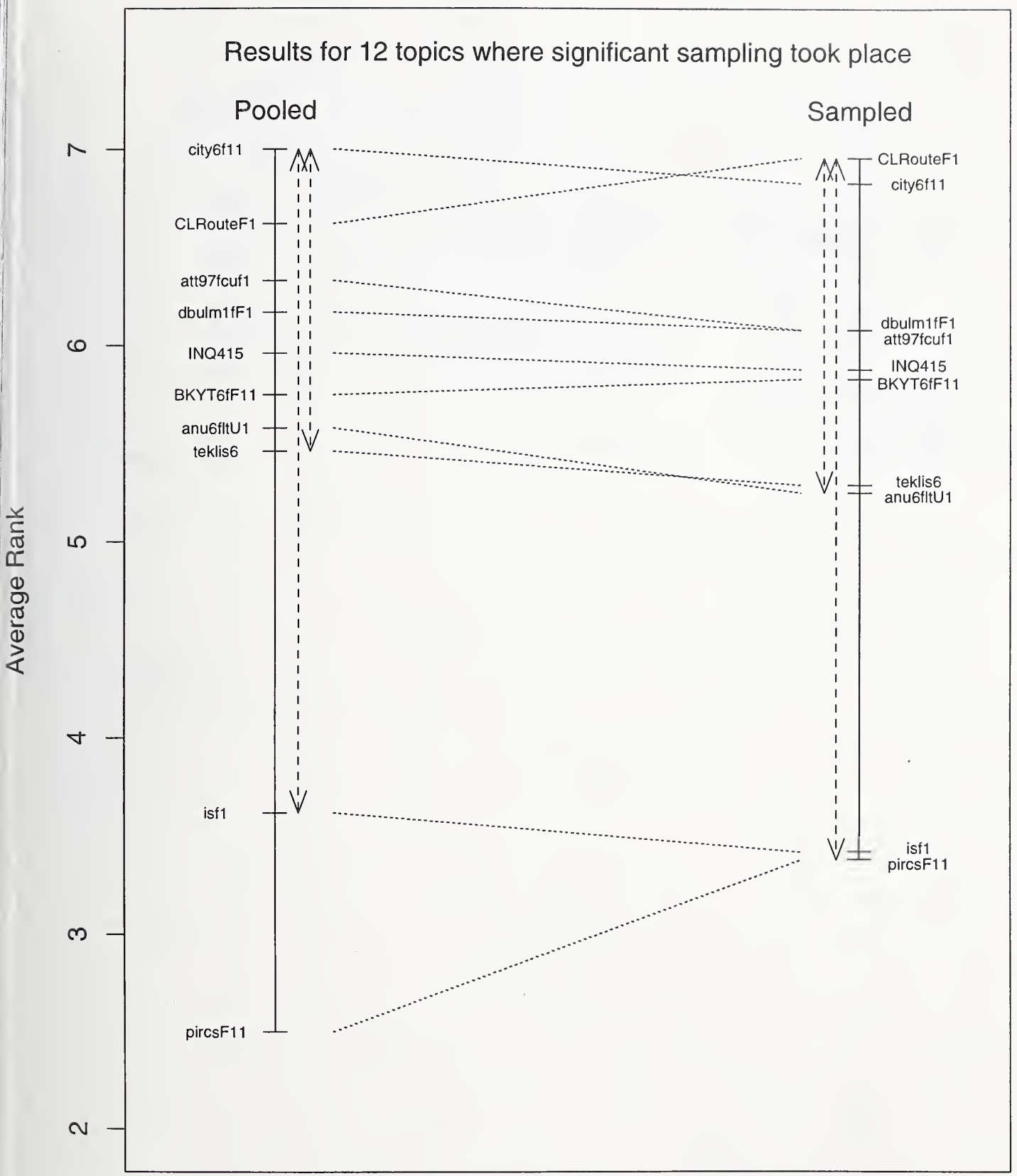


Figure 5 - Observed vs. Optimal F1 Utility

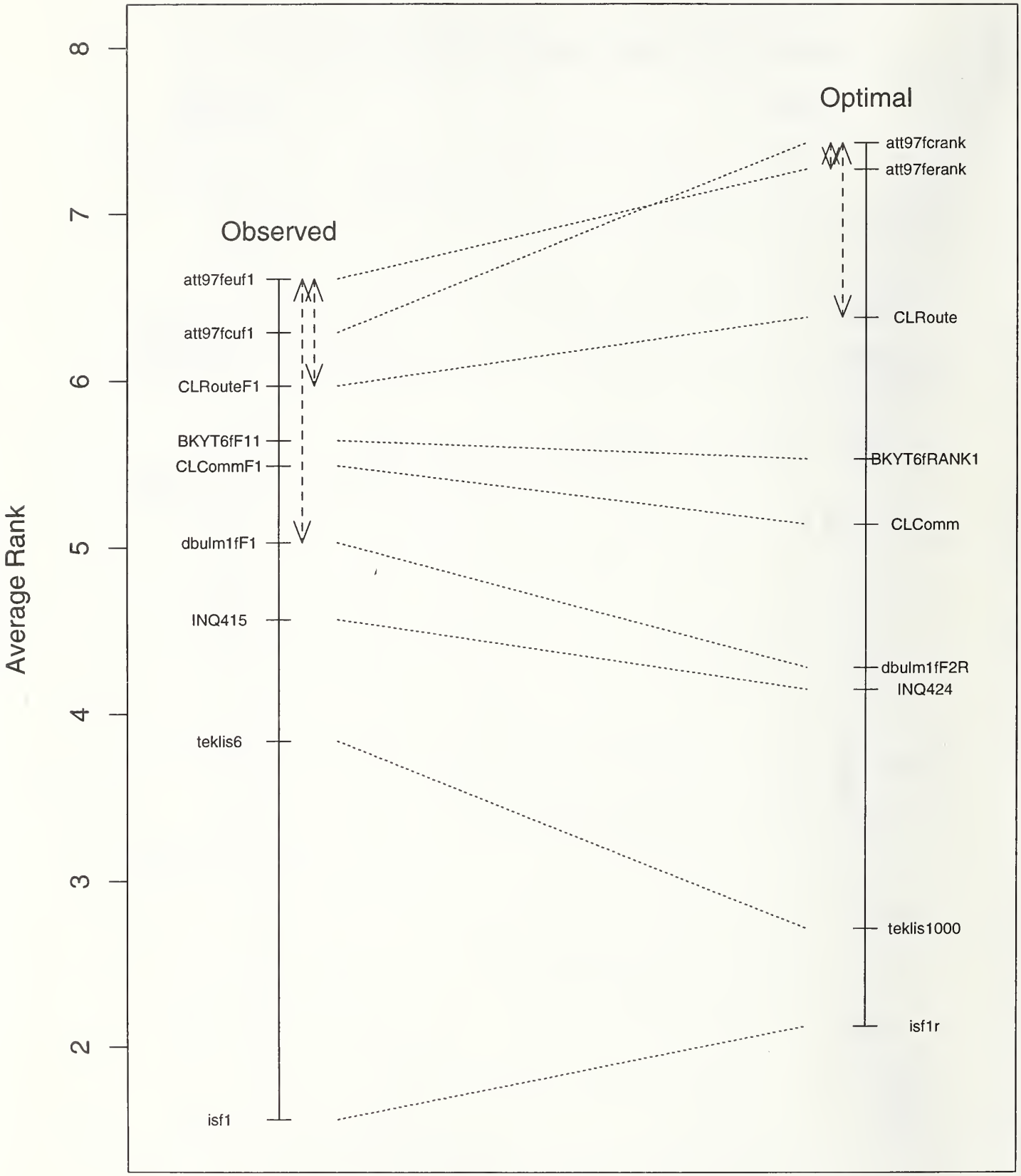


Figure 6 - Old vs. New Topics (F1 Utility)

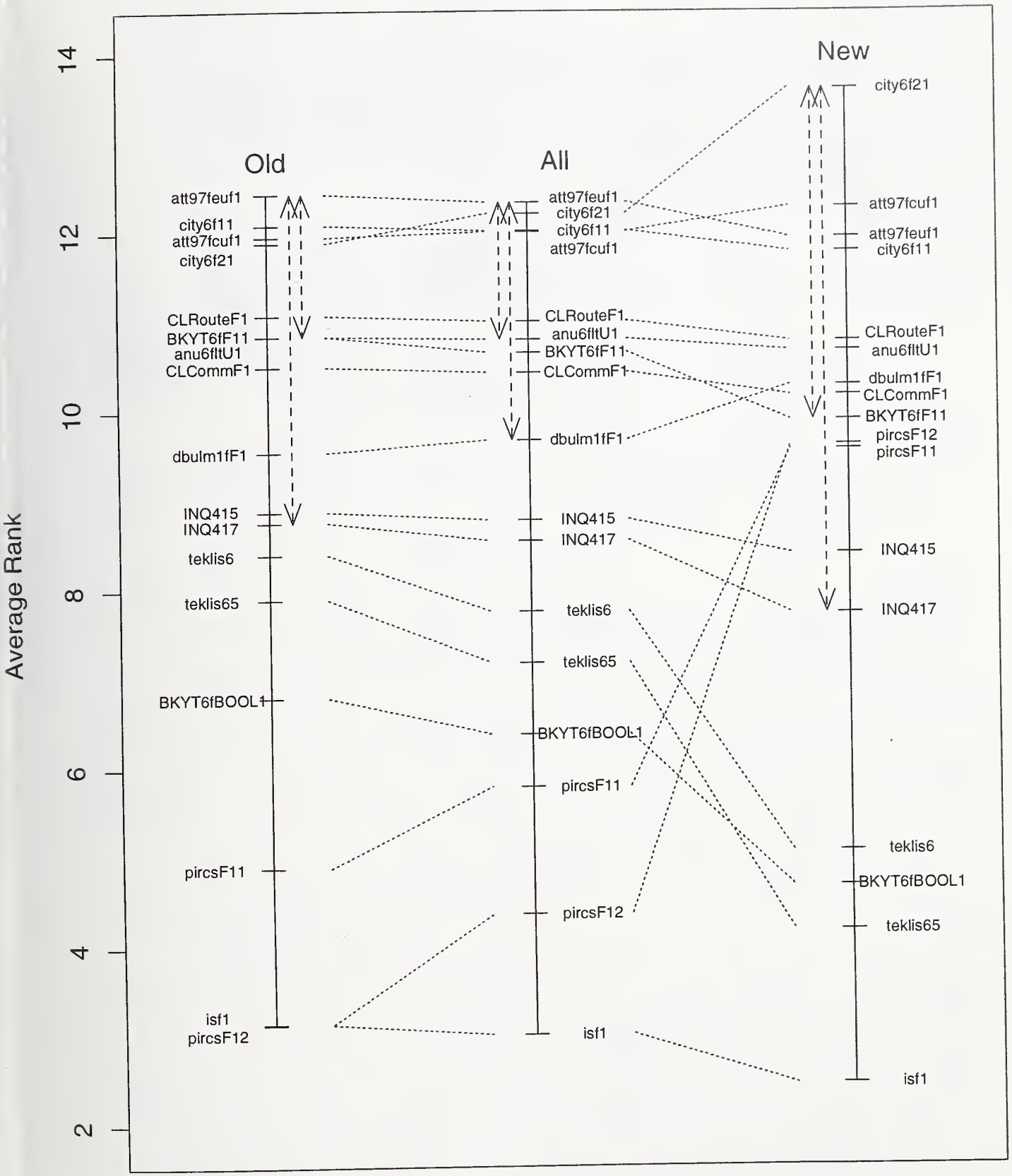


Figure 7 - Easy vs. Hard Topics (F1 Utility)

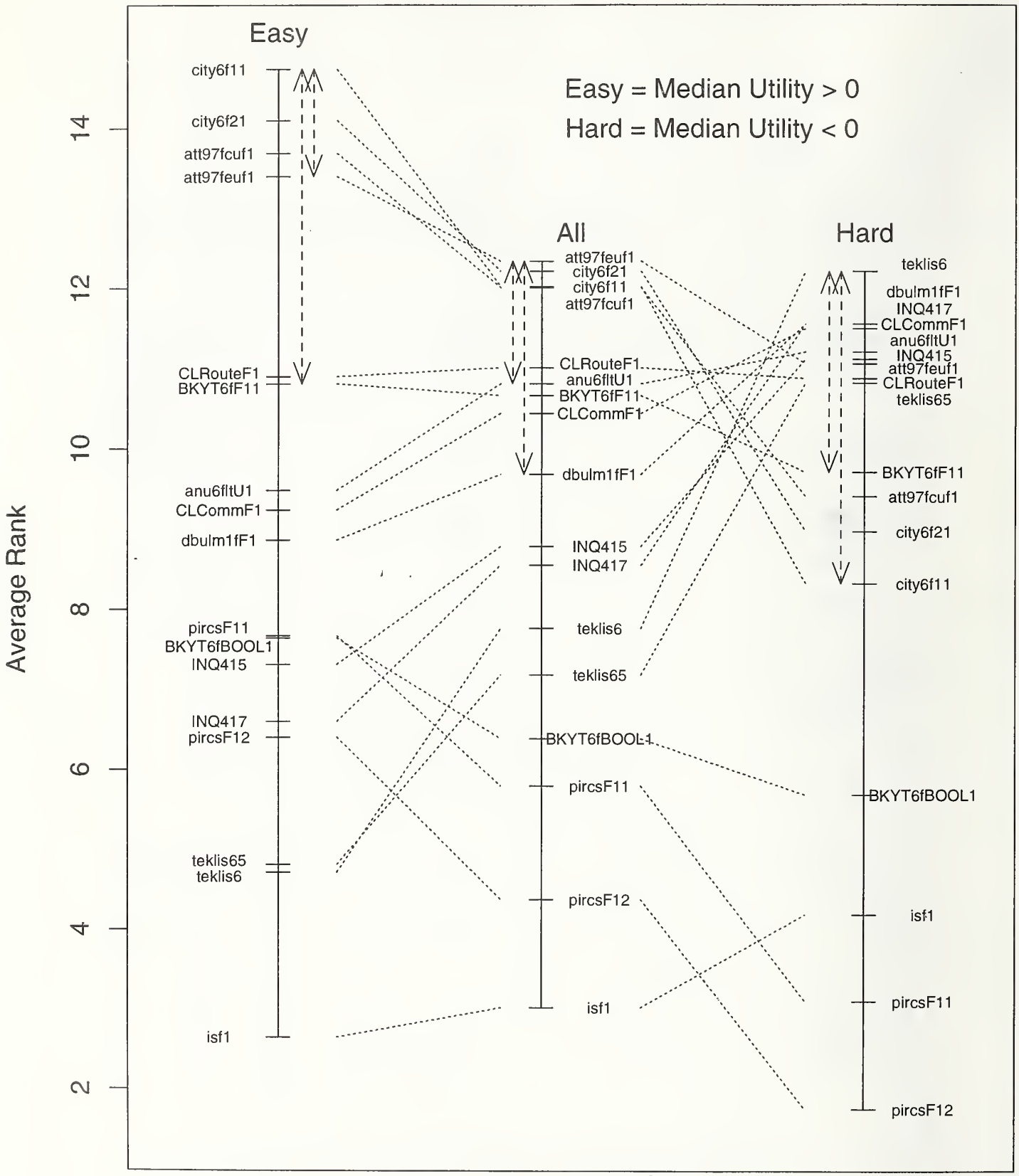
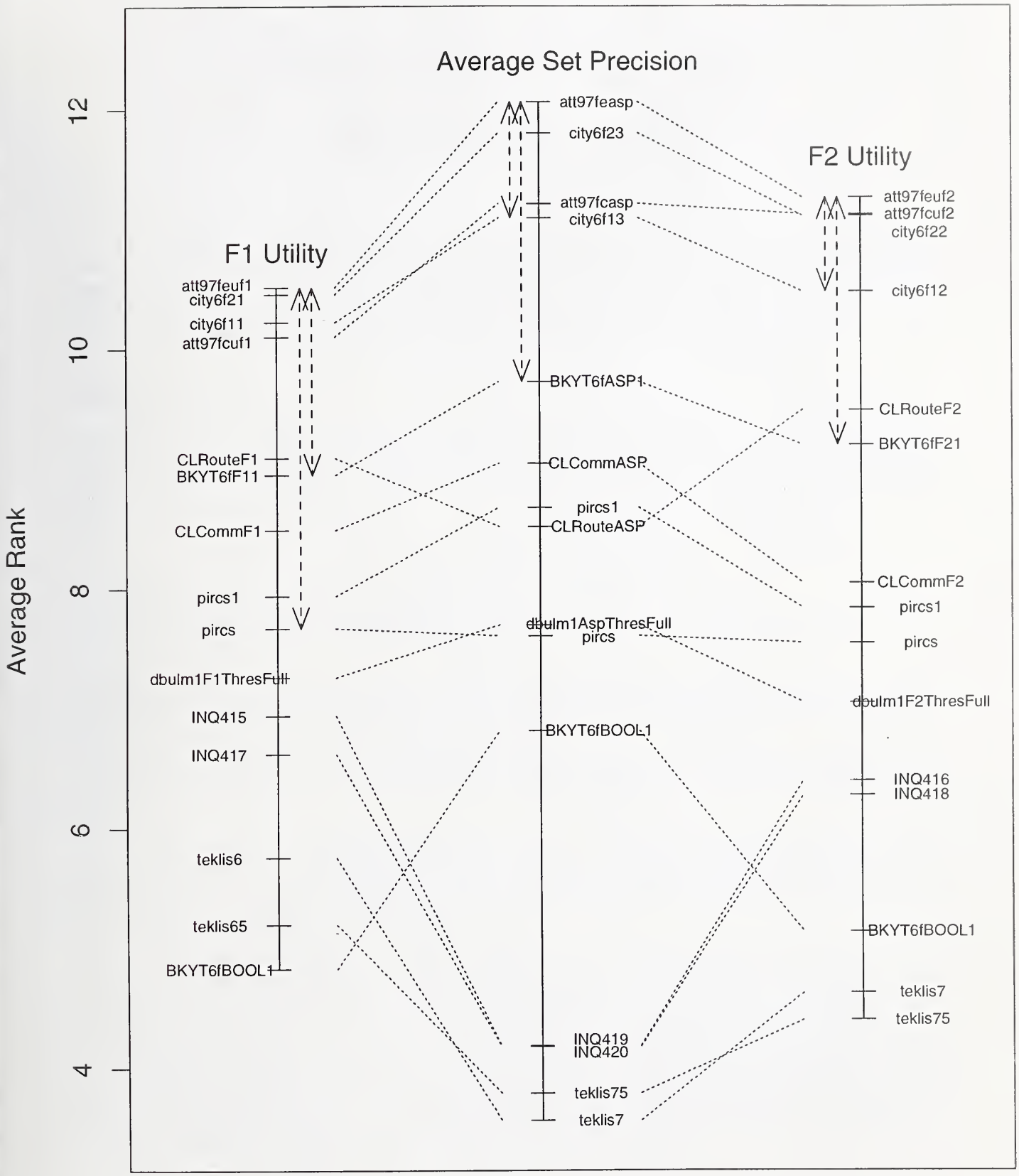


Figure 8 - dbulm and pircs corrected



TREC 6 High-Precision Track

Chris Buckley (SabIR Research Inc)

Abstract

The TREC High-Precision (HP) track compares systems on the simple "Real World" task of having users finding a few relevant documents as quickly as possible. Five groups are participating with each happening to emphasize different aspects of the retrieval process, from visualization to structured queries to relevance feedback. With so few groups participating in this inaugural run of the the track, no decisive conclusions can be reached. However, the indications are that simple approaches work very well.

Track Description

The High-Precision Track is a new track for TREC. It has a very simple short description: for each query, a user should find the best 10 documents possible within 5 minutes clock time. One realistic scenario corresponding to this task might be that your boss asks you for a quick report on some area and you need to find some information on the area fast.

There are no restrictions on the type of resources the user may use during this task other than

- Only one user per query per run (no human collaboration).
- The user and system can have no previous information about the query (eg, the system cannot have previously built a query dependent data structure.)

In particular, the users are allowed to make multiple retrieval runs, allowed to look at documents, allowed to use whatever visualization tools the system has, and allowed to use system or collection-dependent thesauruses, as long as they stay within the 5 minute clock time.

This track tests (at least) the effectiveness, efficiency, and user interface of the systems. The task provides a forum for testing many of the neat ideas in user interface and visualization that have been suggested over the years.

Unlike other interactive evaluations (for example, the TREC 6 Interactive task), no attempt is made to factor out user differences when comparing across systems. All users are assumed to be experts and equally proficient in use of their own system. This allows for fair comparison of systems, but implies that the absolute level of performance within the track will be better than the level obtainable from casual users. These are upper-bound interactive experiments.

The primary evaluation measure is precision at 10 documents. If less than 10 documents were submitted for a query, the missing documents were counted as non-relevant. *Relative precision* at 10 documents is also calculated. Relative precision gives the precision score relative to the maximum precision score possible. For example, if a query only has two relevant documents, then retrieving both in the top 10 will give a precision score of .20, as opposed to a relative precision score of 1.0. Relative precision is good for those evaluation environments where results are averaged over queries; it weights each query equally, as opposed to precision which considers queries with very few relevant documents as less important. A third experimental measure is unranked average precision, roughly corresponding to the average precision evaluations in other tasks (with most relevant documents having a precision of 0 since they are not retrieved).

Run	Precision	Relative Precision	Unranked Avg-Precision
Cor6HP3	.6020	.6298	.1021
otc3	.5800	.6020	.1067
uwmt6h0	.5720	.5977	.0902
uwmt6h1	.5680	.5834	.0982
Cor6HP2	.5660	.5820	.0786
uwmt6h2	.5640	.5951	.0997
otc1	.5500	.5700	.0973
Cor6HP1	.5440	.5564	.0799
otc2	.5440	.5587	.0916
pirc7Ha	.4260	.4384	.0574
pirc7Ht	.3980	.4163	.0766
DCU97HP	.3820	.4031	.0633
pirc7Hd	.3360	.3509	.0561

Table 1: High Precision Results - 50 Queries

In general, once averaging occurs

- Precision is biased against queries with fewer than 10 relevant documents.
- Relative Precision treats all queries equally.
- Unranked average precision strongly favors those queries with few relevant documents.

The same document set and 50 queries are used for the HP track as for the TREC-6 ad-hoc task. Each group could turn in up to 3 runs. NIST put all retrieved HP documents into the ad-hoc judgement pool; this added few documents to the pool since only 10 documents per query were submitted.

Track Results

Five groups submitted runs in the HP track, using a variety of approaches.

1. Cornell/SabIR: Used parts of the original topic as a query, judged documents, and used automatic relevance feedback.
2. Dublin: Visualization of individual documents (Document At A Glance)
3. OpenText: Submitted runs but did not attend conference or describe runs.
4. Queens: Submitted top 10 documents of automatic runs tuned for high precision (no users).
5. Waterloo: Faceted Boolean queries, choice of sets of documents, passages.

Table 1 gives the results for all runs in the HP track. The results are sorted in decreasing order of precision. Comparing the ordering of groups in each column, relative precision gives basically the same ordering as precision. Where there are swaps in ordering (e.g., between Cor6HP2 and uwmt6h2), we know that one system (in this case Cor6HP2) did comparatively poorly on those queries with less than 10 relevant documents. That is confirmed by looking at the third column which even more strongly favors queries with fewer relevant documents. (The last column is so strongly affected by queries with few relevant documents that it is not a reasonable evaluation measure. This measure will not be used in future HP evaluations, though it is valid in other environments where the system determines the number of retrieved documents.)

Given the preliminary nature of this track, there is not too much analysis possible. The Pirc runs were entirely automatic, and as would be expected, did not do well when compared against the other manual

runs. Dublin's document-at-a-glance approach had serious efficiency problems and was done by a naive user. OpenText did well, but unfortunately has not ever described what they did! The Cornell/SabIR and Waterloo groups also did well; but each group thinks they can improve significantly now that they know where the critical points are.

Cornell/SabIR used the same basic approach for all 3 of their runs, as did Waterloo. The differences in the 3 results are due to the individual tweaks that each user added to the basic approach. For example, Cor6HP3 used much smaller initial queries than Cor6HP2 or Cor6HP1. Thus the system was able to finish the good complicated relevance feedback algorithms much more quickly than with the other users, and thus judgements were made on a better set of documents. It is clear that computational efficiency played a big part in the differences between runs.

Conclusion

The overall design of the High-Precision track at TREC 6 worked well. The participants all agreed that it was a good task that could be fairly evaluated and that they all learned substantial amounts about their system, able to improve their system for the TREC 6 task and getting ideas for future improvements. There were not enough full participants to be able to draw any conclusions about the relative value of the basic approaches. The preliminary results indicate that light user interaction (user only judges documents relevant or not) does as well as more complicated user interactions. However, any definitive decision on this remains for future TRECs!

TREC-6 Interactive Track Report

Paul Over
over@nist.gov
Natural Language Processing and Information Retrieval Group
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA

July 29, 1998

Abstract

This report is an introduction to the work of the TREC-6 Interactive Track with its goal of investigating interactive information retrieval by examining the process as well as the results.

Twelve interactive information retrieval (IR) systems were run on a shared problem: a question-answering task, six statements of information need, and a collection of 210,158 articles from the Financial Times of London 1991-1994. The track specification called for two levels of experimentation: cross-site system comparisons in terms of simple measures of end results and local experiments with their own hypotheses and attention to the search process.

This report summarizes the cross-site experiment. It refers the reader to separate discussions of the experiments performed at each participating site - their hypotheses, experimental systems, and results.

The cross-site experiment can be seen as a case study in the application of experimental design principles and the use of a shared control IR system in addressing the problems of comparing experimental interactive IR systems across sites: isolating the effects of topics, human searchers, and other site-specific factors within an affordable design.

The cross-site results confirm the dominance of the topic effect, show the searcher effect is almost as often absent as present, and indicate that for several sites the 2-factor interactions are negligible. An analysis of variance found the system effect to be significant, but a multiple comparisons test found no significant pairwise differences.

1 Introduction

The high-level goal of the TREC-6 Interactive Track was the investigation of searching as an interactive information retrieval (IR) task by examining the process as well as the outcome. To these ends the track specification provided for two levels of experimentation.

One level focused on cross-site system comparison in terms of simple summary measures of end results, treating each of the 12 participating experimental systems as a black box. This report provides a brief introduction to this level – essentially a synopsis of the fuller treatment in Lagergren and Over (to appear in the proceedings of SIGIR'98). Supporting materials and results are included in the results section of these proceedings and are available online (NIST, 1998a).

The other level comprised the experiments carried out at each site, producing data for the system comparison, but at the same time reflecting their own research goals and many different approaches to interactive searching. Readers should consult the site reports in these proceedings for information about the experiments and experimental system(s) run at each site (see fig. 1).

Group	Experimental system(s)	Searchers per system
City University, London	city	8
IBM's T. J. Watson Research Center	IBM	4
New Mexico State Univ. at Las Cruces	NMSU	4
Oregon Health Sciences Univ.	OHSU	4
Royal Melbourne Institute of Technology	rmit	4
Rutgers University	rutint1, rutint2	4
University of California at Berkeley	BrklyINT	4
University of Massachusetts at Amherst	INQ4iai, INQ4iaip	8
University of North Carolina at Chapel Hill	unc6ia, unc6ip	4

Figure 1: Groups, systems, and searchers in the TREC-6 Interactive Track experiment.

2 Motivation for the experimental design

By a combination of choice and necessity, the interactive track for TREC-6 adopted an approach to cross-site system comparison which is significantly different from those taken by the main TREC tasks and the other tracks. The principal difference concerns the control of the main factors, their two-way interactions, and other site-specific effects.

Within the interactive track, a human searcher is always involved and practical limits on available searcher time, a scarce resource for many participating groups, mean that only a small number of topics can be used for each searcher. High experimenter investment per searcher and the interactive track's goal of investigating the process as well as the result of interactive searching underscore the importance of extracting as much information from each experiment as possible. As a result the track participants wanted to measure separately the effect of topics, searchers, and systems as well as gather some information about the strength of expected interactions between system and topic, topic and searcher, and searcher and system. In addition they wanted to eliminate any site-specific effects not due to systems.

Although the topics and the collection were available at all sites, experimental participants could not be randomly assigned to experimental systems. In other words it was not possible to install all systems at one experimental site, provide reliably usable network access to all systems from all sites, or transport one set of experimental participants to all sites.

Out of discussions following TREC-5 emerged a compromise design, which uses a single basic IR system installed as a control at all sites – a common yardstick against which to measure all the experimental systems. The measure of interest was the difference between the performance on an experimental system and performance on the control $(E - C)$ for a given searcher. The basic experimental design, a Latin square, allowed unbiased estimation of how much better the experimental system was than the control – unconfounded by the main effects of topic and searcher. The effect of expected interactions was reduced by replicating the basic Latin square.

3 Method

3.1 Participants

Each of the nine participating groups selected its own participants, known in what follows as "searchers," with only one restriction: no searcher could have previously used either the control system or the experimental system. Additional restrictions were judged impractical given the difficulty of finding searchers. Standard demographic data about each searcher was collected by each site and some sites administered additional tests.

3.2 Apparatus

IR systems

In addition to running its experimental system(s), each participating site installed and ran a simplified version of ZPRISE 2.0, a public domain IR package developed by NIST (NIST, 1998b). The proximity, phrase, and fielded search support in ZPRISE were turned off, as was support for relevance feedback.

Computing resources

Each participating group was responsible for its own computing resources adequate to run both the control and experimental systems and collect the data required for both the matrix and embedded experiments. The control and the experimental systems were to be provided with equal computing resources within a site but not necessarily the same as those provided at other sites.

Topics

Six of the 50 topics created by NIST for the TREC-6 adhoc task were selected and modified for use in the interactive track by adding a section called "Aspects." The six topics were entitled as follows:

- 326i Ferry sinkings
- 322i International art crime
- 307i New hydroelectric projects
- 347i Wildlife extinctions
- 303i Hubble telescope achievements
- 339i Alzheimer's drug treatment

Each of the topics describes an information need with many aspects - an aspect being roughly one of many possible answers to a question which the topic in effect poses. Here is an abbreviated example interactive topic. Note the "Aspects" paragraph.

```
Number: 326i

Title: Ferry Sinkings

Description:

Any report of a ferry sinking where
100 or more people lost their lives.

Narrative:

To be relevant, a document must identify a
ferry that has sunk causing the death of
100 or more humans....

Aspects:

Please save at least one RELEVANT document
that identifies EACH DIFFERENT ferry sinking
of the sort described above. If one document
discusses several such sinkings, then you
need not save other documents that repeat
those aspects, since your goal is to identi-
fy different sinkings of the sort described
above.
```

Searcher task

The task of the interactive searcher was to save relevant documents, which, taken together, covered as many different aspects of the topic as possible in the 20 minutes allowed per search.

Searchers were encouraged to avoid saving documents which contributed no aspects beyond those in documents already saved, but were to be told there was no scoring penalty for doing so.

Document collection

The collection of documents to be searched was the Financial Times of London 1991-1994 collection (part of the TREC-6 adhoc collection). This collection contains 210,158 documents (articles) totaling 564 megabytes. The median number of terms per document is 316 and the mean is 412.7. NIST indexed the collection for use by ZPRISE and distributed the ZPRISE index to participating sites.

3.3 Procedure

Each searcher performed six searches on the collection using the six TREC-6 interactive track topics. The order in which each searcher saw the topics was determined by random draw and was identical for all sites and searchers.

The minimal 4-searcher-by-6-topic matrix was constructed of six 2-searcher-by-2-topic Latin squares. Each 2-by-2 square blocks for the main topic and searcher effects and repetition of the 2-by-2 square

"Site" experimental matrix - as evaluated

Topics ⇒ / Searchers ⇓	326i	347i	322i	303i	307i	339i
1	E	C	E	C	E	C
2	C	E	C	E	C	E
3	E	C	E	C	E	C
4	C	E	C	E	C	E

Figure 2: Minimal 4-searcher-by-6-topic matrix as evaluated. E = experimental system, C = control

"Site" experimental matrix - as run

Topics ⇒ / Searchers ⇓	326i	322i	307i	347i	303i	339i
1	E	E	E	C	C	C
2	C	C	C	E	E	E
3	E	E	E	C	C	C
4	C	C	C	E	E	E

Figure 3: Minimal 4-searcher-by-6-topic matrix as run.

reduces the effect of any remaining interactions. The matrix in Figure 2 was the basis for the evaluation of the results. Each 2-by-2 square yields 2 $E - C$ differences for a total of 12 differences for each 4-searcher-by-6-topic matrix.

To reduce the searcher's cognitive load and possible confusion due to switching search systems with each search, the columns were permuted as indicated in Figure 3 for the running of the experiment.

In resolving experimental design questions not covered here (e.g., scheduling of tutorials and searches, etc.), participating sites were asked to minimize the differences between the conditions under which a given searcher used the control and those under which he or she used the experimental system.

3.4 Data submitted to NIST for evaluation

Four sorts of result data were collected for evaluation/analysis (for all searches unless otherwise specified) and are available from the TREC-6 Interactive Track web page (NIST, 1998a).

- sparse-format data - list of documents saved and the elapsed clock time for each search
- rich-format data - searcher input and significant events in the course of the interaction and their timing
- a full narrative description of one interactive session for topic 326i
- any further guidance or refinement of the task specification given to the searchers

Only the sparse format data were evaluated at NIST to produce a triple for each search: aspectual precision and recall (these as defined in the next section) and elapsed clock time.

3.5 Evaluation of data submitted to NIST

Evaluation by NIST of the sparse-format data proceeded as follows. For each topic, a pool was formed

containing the unique documents saved by at least one searcher for that topic regardless of site.

For each topic, the NIST assessor, normally the topic author, was asked to:

1. Read the topic carefully.
2. Read each of the documents from the pool for that topic and gradually:
   (a) Create a list of the aspects found somewhere in the documents
   (b) Select and record a short phrase describing each aspect found
   (c) Determine which documents contain which aspects
   (d) Bracket each aspect in the text of the document in which it was found

Then for each search (by a given searcher for a given topic at a given site), NIST used the submitted list of selected documents and the assessor's aspect-document mapping for the topic to calculate:

- the fraction of total aspects (as determined by the assessor) for the topic that are covered by the submitted documents (i.e., aspectual recall)
- the fraction of the submitted documents which contain one or more aspects (i.e., aspectual precision)

The third measure, elapsed clock time, was taken directly from the submitted results for each search.

4 Results

4.1 Main results

The analysis proceeded in two stages:

- analysis of the data from each site independently to determine how best to model its data in terms of the main effects and interactions of interest to the track participants
- combination and analysis of the data across sites to yield the desired cross-site system comparison

The "treatment effect" discussed is the difference between the aspectual recall of the experimental and control systems ($E - C$). Only the analysis for recall is presented here since the interactive track task was seen by participating groups primarily as a recall-oriented problem and the recall data are more precise than the precision data. Of the 13 sets of results submitted, 10 were in the correct format for cross-site comparison.

Separate analysis for each site

For each site we considered the following four models for $y(i,j,k) =$:

(M1) $m + s(i) + t(j) + p(k) + e(i,j,k)$

(M2) $m + s(i) + t(j) + p(k) + ST(i,j) + e(i,j,k)$

(M3) $m + s(i) + t(j) + p(k) + SP(i,k) + e(i,j,k)$

(M4) $m + s(i) + t(j) + p(k) + ST(i,j) + SP(i,k) + e(i,j,k)$

where

$y(i,j,k)$ = recall for system i, topic j, searcher k

m = the mean recall for the site

$s(i)$ = effect of system i, where $i = 1$ (C), 2 (E)

$t(j)$ = effect of topic j, where $j = 1$ to 6 topics

$p(k)$ = effect of searcher k where $k = 1$ to 4 or 8 searchers

$ST(i,j)$ = interaction between system i and topic j; NOTE: this is not the product of $s(i)$ and $t(j)$

$SP(i,k)$ = interaction between system i and searcher k; NOTE: this is not the product of $s(i)$ and $p(k)$

$e(i,j,k)$ = the random error for observation $y(i,j,k)$

The effect $s(i)$ is considered to be a *fixed* effect, that is, an effect for which we are interested in comparing its specific levels, here E versus C (Neter, Wasserman, & Kutner, 1990). The effects $t(j)$ and $p(k)$ are considered to be *random* effects. Random effects are

Table 1: Details on each site's best model for aspectual recall

Site/system	n	E	C	E-C	s(topic)	s(searcher)	s(system* topic)	s(system* searcher)	s(residuals)	s(E-C)	df	t	U	Lower 95% CI limit	Upper 95% CI limit
BrklyINT	24	0.5725	0.4937	0.079	0.325	0.000	0.067	0.057	0.081	0.065	2	4.30	0.279	-0.200	0.358
IBM	24	0.2638	0.3778	-0.114	0.195	0.000	0.153	-	0.149	0.107	4	2.78	0.297	-0.411	0.183
INQ4iai	48	0.3645	0.4511	-0.087	0.277	0.091	-	0.049	0.133	0.046	6	2.45	0.112	-0.198	0.025
INQ4iaip	48	0.4995	0.4380	0.062	0.339	0.046	0.066	-	0.103	0.048	4	2.78	0.133	-0.072	0.195
NMSU	24	0.4719	0.4523	0.020	0.337	0.076	-	-	0.061	0.025	14	2.14	0.053	-0.034	0.073
OHSU	24	0.3730	0.4901	-0.117	0.295	0.000	0.118	-	0.109	0.081	4	2.78	0.226	-0.343	0.109
city	48	0.4000	0.3810	0.019	0.267	0.070	-	-	0.167	0.048	34	2.03	0.098	-0.079	0.117
rmit	24	0.4663	0.4993	-0.033	0.279	0.093	0.026	0.040	0.078	0.045	2	4.30	0.195	-0.228	0.162
unc6ia	24	0.4441	0.5113	-0.067	0.312	0.000	0.073	-	0.142	0.072	4	2.78	0.199	-0.266	0.132
unc6ip	24	0.4666	0.4551	0.012	0.340	0.090	-	-	0.119	0.049	14	2.14	0.104	-0.093	0.116

effects for which we are not interested in comparing their specific levels, but rather choose the levels to be a random or representative sample from some population of interest. Interactions involving random effects are also treated as random effects, so $ST(i,j)$ and $SP(i,k)$ are treated as random effects. The random error term $e(i,j,k)$ is always treated as a random effect. Random effects are typically assumed to be normally distributed with mean zero and given variance. We write these assumptions as

$$\begin{aligned} t(j) &\sim N(0,\sigma_t^2) \\ p(k) &\sim N(0,\sigma_p^2) \\ ST(i,j) &\sim N(0,\sigma_{ST}^2) \\ SP(i,k) &\sim N(0,\sigma_{SP}^2) \\ e(i,j,k) &\sim N(0,\sigma_e^2) \end{aligned}$$

where "$\sim N(\mu,\sigma^2)$" means "is normally distributed with mean μ and variance σ^2". From these assumptions we observe, for example, that the variance of $y(i,j,k)$ for model (M4) is not σ_e^2 as it would be for a pure fixed effects model, but rather

$$\sigma_t^2 + \sigma_p^2 + \sigma_{ST}^2 + \sigma_{SP}^2 + \sigma_e^2$$

Since the variance of the random effects partition the variance of y, they are called variance components. The presence of random effects also implies that the $y(i,j,k)$'s are not independent for a given system. This is easily seen by the fact that recall will tend to be higher for easier topics than for more challenging topics.

Models that include both fixed and random effects (apart from the random error term) are called *mixed* models. SAS's Proc MIXED (Littell, Milliken, Stroup, & Wolfinger, 1996) estimates parameters in a mixed model. Proc MIXED was used here to estimate the parameters in each of the four models for each site. The best model for each site was then selected based on residual plots and significance testing. The results for the best models are given in Table 1 where

n is the number of observations

E is the mean of the experimental system data

C is the mean of the control system data

$s(topic)$ estimates σ_t

$s(searcher)$ estimates σ_p

$s(system * topic)$ estimates σ_{ST}

$s(system * searcher)$ estimates σ_{SP}

$s(residuals)$ estimates σ_e

$s(E-C)$ estimates the standard deviation of $E-C$

df is the degrees of freedom for $s(E-C)$

t is the t-value with df degrees of freedom for a 95% confidence interval

$U = t * s(E-C)$ is the 95% uncertainty for $E-C$

$Lower\ 95\%\ CI\ limit = (E-C) - U$

$Upper\ 95\%\ CI\ limit = (E-C) + U$

A missing standard deviation estimate ("-") indicates that it is negligible.

The following observations about Table 1 are worth noting:

1. $s(topic)$ is the largest standard deviation for each site. So running the replicated Latin square design, which eliminated the topic (and searcher) effect from comparisons of E and C, was crucial.

2. For 4 of 10 sites, the searcher effect was negligible.

3. Model (M1) was best for 3 sites, model (M2) for 4 sites, model (M3) for 1 site, and model (M4) for 2 sites.

4. Since the confidence intervals for the true $E-C$ (see last two columns of Table 1) contain zero for each site, one would not conclude that E differs from C for any site.

5. For 5 of the 7 cases where interactions are present in the model, their standard deviation is less than the standard deviation for the error term.

Cross-site analysis

A cross-site analysis of variance showed the site factor was statistically significant, since the p-value for the ANOVA F test was $0.0133 < \alpha = 0.05$. This indicates that the mean $E-C$ differed across sites.

The next step was to determine for which specific sites, the mean $E-C$'s differ using multiple comparisons. Several techniques are available for multiple comparisons. Since pairwise differences were of primary interest, Tukey's Studentized Range Test ($\alpha = 0.05$) was used, adjusted for unequal sample sizes. It indicated that none of the pairs contained means that were statistically different. While this seems surprising, the significance of the ANOVA F test does not guarantee that a pairwise difference will be statistically significant. While Tukey's test is more powerful than Scheffé's, it is generally less powerful than the F test.

5 Discussion

5.1 General findings

Although the cross-site comparison did not quite detect differences between systems with the current design, the cross-site and within-site analyses provide thought-provoking information on variability, sizes of main effects, and presence/absence of 2-way interactions that can be used to design improved experiments more likely to detect any such differences.

The results confirm the importance of applying good experimental design principles to extract maximal information from interactive IR experiments while minimizing their cost. For example, since the topic effect was dominant, good experiment design was critical for eliminating its effect from system comparisons.

The lack of a strong searcher effect for almost half of the sites was surprising to us, as was, to a lesser degree, the weakness or absence of searcher-topic and searcher-system interactions. Would other sets of systems, searchers, and/or topics yield similar findings?

Finally, the results suggest that reasonably precise pairwise comparisons of systems are possible using more searchers.

5.2 Future research

Questions which remain to be addressed include the following. Two concern the analysis of existing results and two pertain to possible future experiments.

- The TREC-6 Interactive Track cross-site experimental design *assumes* that the control is effective in eliminating site-related effects. Outside the bounds of the experiment, this assumption was tested in a pre-experiment at three sites (see NIST, 1998a) and by additional experiments performed by the team at the University of Massachusetts (UMass) before TREC-6. All of these experiments contrasted direct comparison of two experimental systems with indirect comparison (via the control). In general the two methods produced surprisingly different results. However, due to large underlying variability, the estimates produced by the two methods were not statistically different. (Note, however, that Swan and Allan (to appear in thé proceedings of SIGIR'98) also evaluated the effectiveness of the control and, using using data from 24 additional direct-comparison searches, draw a clearly negative conclusion.)

  In any case, for practical purposes the use of the control as described cannot be recommended, because its high cost can only be justified on the basis of positive evidence for its effectiveness and several attempts have failed to produce such evidence. The reasons for this lack of positive evidence deserve further study.

- How, if at all, are the data collected by some sites on the characteristics of the searchers related to the searchers' performance?

- How do the aspects identified by the searchers and the assessors compare? What, if anything, does their (dis)agreement tell us about the consistency with which the task was understood and executed across sites? What are the consequences of this (in)consistency for the variability of the dependent variable?

- If the experiment were to be re-run, should the searcher task be simplified to reduce the cognitive load and perhaps decrease variability of results by eliminating relevance of documents as a consideration for searchers and assessors - making the task just question-answering?

- Would it be feasible to eliminate the use of a common control by comparing multiple *experimental* systems per site, e.g., site A's $E1$ and site B's $E2$ at site A and site B's $E2$ and site C's $E3$ at site B, etc., thus reducing the number of runs needed to achieve a desired uncertainty?

6 Author's note

The design of the TREC-6 Interactive Track matrix experiment grew out of the efforts of the many people who contributed to the discussion of ends and means on the track discussion list and through other channels. The author would like to acknowledge the contributions of the track coordinators, Steve Robertson and Nick Belkin as well as those of Peter Pirolli and others (then) at Xerox PARC. Special thanks go to Eric Lagergren of NIST's Statistical Engineering Division for his guidance in the design and interpretation of the experiment and for performing the analysis of the summary data.

References

Lagergren, E. and Over, P. (to appear in the proceedings of SIGIR'98). *Comparing Interactive Information Retrieval Systems Across Sites: the TREC-6 Interactive Track Matrix Experiment.*

Littell, R., Milliken, G., Stroup, W., and Wolfinger, R. (1996). *SAS System for Mixed Models.* Cary, NC, USA: SAS Institute.

Neter, J., Wasserman, W., and Kutner, M. (1990). *Applied Linear Statistical Models.* Boston, MA, USA: Irwin.

NIST. (1998a). *TREC-6 Interactive Track Home Page* [URL]. www-nlpir.nist.gov/~over/t6i.

NIST. (1998b). *The ZPRISE 2.0 Home Page* [URL]. www-nlpir.nist.gov/~over/zp2.

Swan, R. C. and Allan, J. (to appear in the proceedings of SIGIR'98). *Aspect Windows, 3-D Visualizations, and Indirect Comparisons of Information Retrieval Systems.*

7 Appendix: Instructions to be given to each searcher

The following introductory instructions are to be given once to each searcher before the first search:

> Imagine that you have just returned from a visit to your doctor during which it was discovered that you are suffering from high blood pressure. The doctor suggests that you take a new experimental drug, but you wonder what alternative treatments are currently available. You decide to investigate the literature on your own to learn what different alternatives are available to you for high blood pressure treatment. You really need only one document for each of the different treatments for high blood pressure.
>
> You find and save a single document that lists 4 treatment drugs. Then you find and save another 4 documents that each discusses a separate alternative treatment: one that discusses the use of calcium, one that talks about regular exercise, another that mentions biofeedback, and one that cites the snakeroot plant as a possible alternative treatment. In all, you have identified 8 different aspects for this topic in 5 documents.
>
> Now we would like you to identify as many aspects as possible for each topic that will be presented to you. You will be given 20 minutes to search for each topic's aspects. Please save 1 relevant document for each of the aspects that you identify. If you save 1 document that contains many aspects, try not to save additional documents that contain only those aspects, unless a document contains additional aspects as well.
>
> As you identify an aspect, please write down a word or short phrase to identify the aspect - enough to help you keep track of which aspects you have found.
>
> Carefully read each description and narrative for each topic since they provide information on which documents are relevant and because the interpretation of "aspects" changes from topic to topic. For example, aspects can refer to different developments in a field, to different instances in which an event can occur, or to different kinds of treatments, to names of persons, places or things, etc. – as it did in our example above.
>
> Do you have any questions about
>
> - what we mean by aspects
> - what we mean by relevant
> - the way in which you are save nonredundant documents for each aspect

TREC-6 1997 Spoken Document Retrieval Track Overview and Results

John S. Garofolo, Ellen M. Voorhees, Vincent M. Stanford

National Institute of Standards and Technology (NIST)
Information Technology Laboratory
Building 225, Room A-216
Gaithersburg, MD 20899

Karen Sparck Jones

Cambridge University
Cambridge CB2 3QG, U.K.

ABSTRACT

This paper describes the 1997 TREC-6 Spoken Document Retrieval (SDR) Track which implemented a first evaluation of retrieval of broadcast news excerpts using a combination of automatic speech recognition and information retrieval technologies. The motivations behind the SDR Track and background regarding its development and implementation are discussed. The SDR evaluation collection and topics are described and summaries and analyses of the results of the track are presented. Finally, plans for future SDR tracks are described.

Since this was the first implementation of an evaluation of SDR, the evaluation itself as well as the evaluated technology should be considered experimental. The results of the first SDR Track were very encouraging and showed us that SDR could be successfully implemented and evaluated. However, the results of the SDR Track should be considered preliminary since the 50-hour spoken document collection used was very small for retrieval experiments (even though it was considered extremely large for speech recognition purposes.) Nonetheless, with thirteen groups participating in the TREC-6 SDR Track, a considerable amount of experience was gained in implementing and evaluating the SDR task. This experience will greatly benefit the next 1998 TREC-7 SDR Track.

1. MOTIVATION

Spoken Document Retrieval (SDR) involves the retrieval of excerpts from recordings of speech using a combination of automatic speech recognition and information retrieval techniques. In performing SDR, a speech recognition engine is applied to an audio input stream and generates a time-marked textual representation (transcription) of the speech. The transcription is then indexed and may be searched using an Information Retrieval engine. In traditional Information Retrieval, a topic (or query) results in a rank-ordered list of documents. In SDR, a topic results in a rank-ordered list of temporal pointers to relevant excerpts. In an operational SDR system, these excerpts could be topical sections of a recording of a conference or radio or television broadcasts. This technology when mature will permit users to search large collections of non-textual multi-media materials.

SDR was chosen as a TREC-6 (NIST Text REtrieval Conference 6) task for 1997 because of its potential use in navigating large multi-media collections of the near future and because it was believed that the component Speech Recognition and Information Retrieval technologies might work well enough now for usable SDR in some domains. SDR also provides a rich research domain in that it supports both development of large-scale near-real-time continuous speech recognition technologies and technologies for retrieval of spoken language. Further, SDR provides a venue for the development of synergy between the speech recognition and information retrieval communities to improve both technologies and create hybrids.

2. BACKGROUND

In November 1996 at the TREC-5 Workshop and later at the February 1997 DARPA Speech Recognition Workshop, NIST and Karen Sparck-Jones of Cambridge University held a discussion and a call for participation in a Spoken Document Retrieval (SDR) TREC Track for TREC-6. An SDR track would focus research on solving problems inherent in the retrieval of documents created by speech recognition technologies and in the recognition of large quantities of speech. Furthermore, the evaluation component of the track would permit the benchmarking of progress in the retrieval of documents corrupted by recognition errors.

It was decided that the track would involve retrieval of radio and television broadcast news recordings collected by the Linguistic Data Consortium (LDC) in 1996. The LDC Broadcast News (BN) corpus had been collected to support the DARPA-sponsored *Hub-4* continuous speech recognition project and was fully transcribed and annotated with story boundaries and could be adapted at little cost to the SDR task.[1] Both the CSR and IR communities expressed interest in the proposed project, so NIST and Sparck-Jones developed an evaluation plan to implement an initial SDR evaluation during the summer of 1997 to be reported at the November 1997 TREC-6 Workshop.

3. SDR EVALUATION PLAN

Initial discussion involved the nature of the retrieval task to be used in the SDR Track. It was acknowledged that because the amount of available Hub-4 Broadcast News corpora was limited and because this was to be a relatively low-overhead task, a full Ad-hoc-style TREC task was impractical. So, instead, a *known-item* task, which simulates a user seeking a particular, half-remembered document in a collection, was chosen. The goal in a known-item retrieval task is to generate a single correct document for each topic rather than a set of relevant documents as in an ad-hoc task. This approach simplified the selection of topics and eliminated the need for expensive relevance assessments. A known-item retrieval task had been successfully implemented in the similarly-designed TREC-5 *OCR Confusion Track.* [2]

The TREC-6 1997 SDR Evaluation Plan can be found at

http://www.nist.gov/speech/sdr97.txt

3.1 Evaluation Modes

The focus of the initial SDR evaluation was to encourage broad participation from both the Speech Recognition and Information Retrieval Communities. Therefore, the evaluation plan was designed to allow relatively easy entry for members of both communities. Speech recognition and retrieval experts were encouraged to team up to create hybrid SDR systems. The SDR Track included three retrieval conditions which provided control experiments as well as allowing sites without access to speech recognition technology to participate: [3]

Reference (S1) (required) – Retrieval using the "perfect" human-transcribed reference transcriptions of the Broadcast News recordings. This condition provided a control for retrieval.

Baseline (B1) (required) – Retrieval using the IBM-provided speech-recognition-system-generated 1-best transcriptions of the Broadcast News recordings. This condition provided a control for recognition and permitted sites without access to recognition technology to participate.

Full SDR (R1) (optional) – Retrieval using the Broadcast News recordings. This condition required both speech recognition and retrieval (which could be implemented by different sites).

Participants in the Full SDR condition with 1-best word-based recognizers were encouraged to submit the output of their recognition systems to be informally scored by NIST in evaluating the effect of recognition error rates on performance.

For purposes of simplifying the implementation and evaluation process, the hand-annotated temporal story boundaries were given in all conditions. Figure 1. Shows the SDR process for the TREC-6 task.

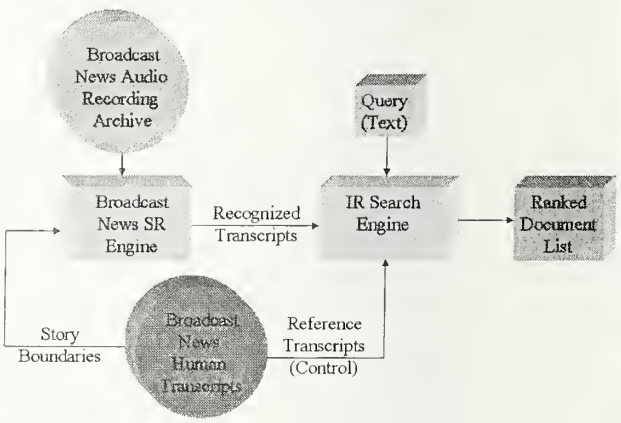


Figure 1. TREC-6 SDR Process.

3.2 Test Corpora

The LDC Broadcast News corpus was chosen for the SDR task since it contained news data from several radio and television sources and was fully transcribed and pre-segmented by story.[1] To adapt the BN corpus to the SDR task, Story ID tags were added to uniquely identify each annotated story.

The existing 100 hours of broadcast news (which was originally collected by the LDC to provide training material for DARPA Hub-4 speech recognition systems)

was divided into equal training and test sets. The 50-hour subset which was used for Hub-4 training in 1996 was chosen for SDR training and the newly-transcribed 50-hour subset was chosen as the test set for the SDR track. This facilitated speech recognition site participation since sites with 1996 Hub-4 systems could apply them directly to the SDR task. [4] (Note that the two sets overlap temporally.)

An index was developed for the 50-hour test set to exclude commercials, sports summaries, weather reports, and untranscribed stories from the test. The baseline recognizer also had bad output for some sections of a few recordings. So that the Baseline test results could be directly compared to those for the Reference and Full SDR conditions, these stories were also removed from the test index.

The final filtered test set contained 1,451 stories with about 400,000 words. About 1/3 of the stories in the test set were labeled as "filler" – non-topical sections of the broadcasts. Because of the small size of the collection for retrieval testing, these were not removed from the test set. The mean length in words for the stories in the test set was 276 words. The histogram in Figure 2 shows the distribution of the length of the stories in the test set.

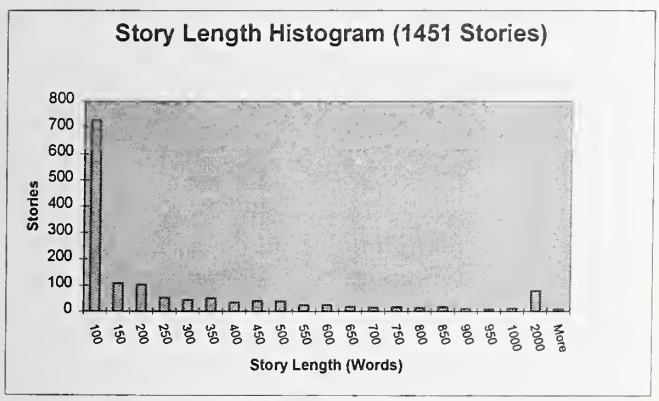


Figure 2. Test Set Story Length Histogram.

Note that about half of the stories contain less than 100 words and a few stories contain 2000 or more words.

The recorded waveform material for the full SDR retrieval mode was made available to the participants in February, 1997. The human-created reference transcripts for the test collection and indices which specified the 1,451 usable stories were released in June. The test topics and baseline recognizer transcripts were released in the beginning of July and results were due at NIST in early September. The results of the SDR track were reported at TREC-6 in November 1997 and at the DARPA Broadcast News Transcription and Understanding Workshop in February 1998.

3.3 SDR Topics

As indicated in section 2.1, a known item retrieval task was selected for the SDR Track. Fifty known item topics, each intended to retrieve a single spoken document, were selected at NIST – half to exercise the retrieval challenges of the task and half to exercise the speech recognition challenges of the task.

To this end, 25 topics were selected by the NIST Spoken Natural Language Processing and Information Retrieval Group to pose various challenges to the retrieval systems. This topic subset is referred to later in this paper as **"Difficult Topics."** An example "difficult" topic (SDR3) was: *What is the difference between the old style classic cinemas and the new styles of cinema we have today?* This topic targeted a story (CNN Headline News, June 7 1996, story 28) which did not contain the word, "cinema." Instead, the document contained several instances of the synonym, "theater." So, this query required systems to use synonymy to retrieve the target story.

The remaining 25 topics were selected by the NIST Spoken Natural Language Processing Group to cover the spectrum of the speech recognition challenges of the task. and divided into two subcategories to emphasize two Hub-4 speech recognition conditions:

> **"Easy" Speech (Hub-4 F0):** High fidelity recording, non-spontaneous speech, native speaker of American English, quiet conditions. An example "F0" topic (SDR1) was: *Does the Olympic torch ever travel by motorcycle?* This topic targeted a story with a scripted reading by a news anchor under low noise/high bandwidth conditions. This story (NPR All Things Considered, June 18 1996, story 9) contained none of the Hub-4-categorized phenomena thought to cause recognition difficulty.
>
> **"Difficult" Speech (Hub-4 FX):** Combinations of speech recognition degrading conditions such as low fidelity channel, spontaneous speech, non-native speaker, noisy conditions. An example "FX" topic (SDR33) was: *In what country do parents fear that the devil is going to come and take their children?* This topic targeted a story (with "medium" fidelity, "high" background noise, and several areas of non-English-speaking speakers with interpreters. Ninety two percent of

the material in the story (CNN Headline News, June 7 1996, story 12) included 2 or more Hub-4-categorized phenomena thought to cause recognition difficulty.

Each of these topics was selected to target at story which contained primarily either Easy (F0) or Difficult (FX) categorized speech. These topic subsets are referred to later in this paper as "Easy Speech" and "Difficult Speech."

One of the 50 topics was removed from the test because it retrieved a story which was in an errorful set of output produced by the baseline recognizer. So, the retrieval for 49 topics was scored and tabulated in the SDR Appendix of the TREC-6 notebook. It was also discovered that two topics properly retrieved multiple stories from the test set. Since this was a known-item task, for simplicity, these were removed in the analyses provided in this paper. Therefore, the results presented in this paper are based on only 47 topics.

4. EVALUATION RESULTS

In all, 13 sites (or site combinations) participated in the SDR Track. Nine of these performed the speech recognition portion as well as retrieval portions of the task and implemented the Reference, Baseline, and Full SDR retrieval conditions:

- AT&T
- Carnegie Mellon University Informedia Group
- Claritech (with CMU Speech Recognition)
- ETH Zurich
- Glasgow (with Sheffield University Speech Recognition)
- IBM
- Royal Melbourne Institute of Technology
- Sheffield University
- University of Massachusetts (with Dragon Systems Speech Recognition)

The remaining 4 sites implemented only the Reference and Baseline retrieval conditions:

- City University of London
- Dublin City University
- University of Maryland
- NSA

4.1 Speech Recognition Component Performance

The primary purpose of the SDR Track was to evaluate the retrieval of spoken documents and not speech recognition. To this end, there was no formal evaluation of the speech recognition component of the Full SDR systems. However, if sites used 1-best word recognition to produce retrieval transcripts for Full SDR, they were encouraged to submit these so that NIST could exam the relationship between word error rate and retrieval performance. Of the eight participating Full SDR sites, four submitted recognition output to NIST for scoring (one of these was the IBM-contributed baseline recognizer.) Other Full SDR sites either used another site's recognizer, used an alternative recognition technique such as phone recognition, or choose not to share their recognition results. Figure 3 shows a histogram of the story Word Error for the IBM system.

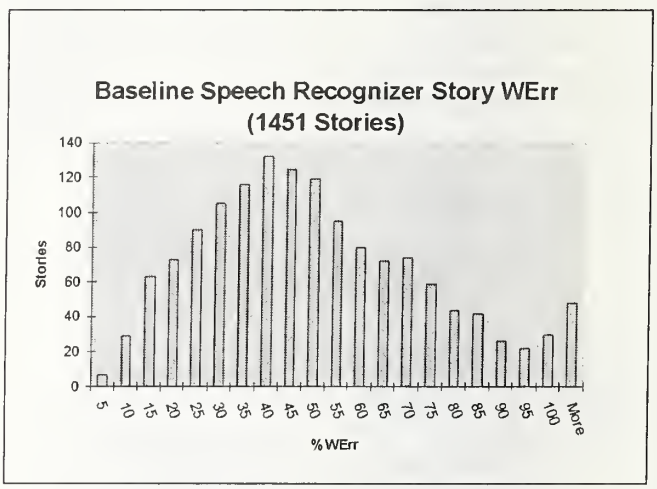


Figure 3. Story Word Error Rate for the IBM Recognizer.

The story Word Error Rate mode for the baseline recognizer was approximately 40% while the mean was substantially higher at 50.0% because of the long right tail in the distribution. The mean story Word Error Rate for the other recognizers fell between approximately 35% and 40%.

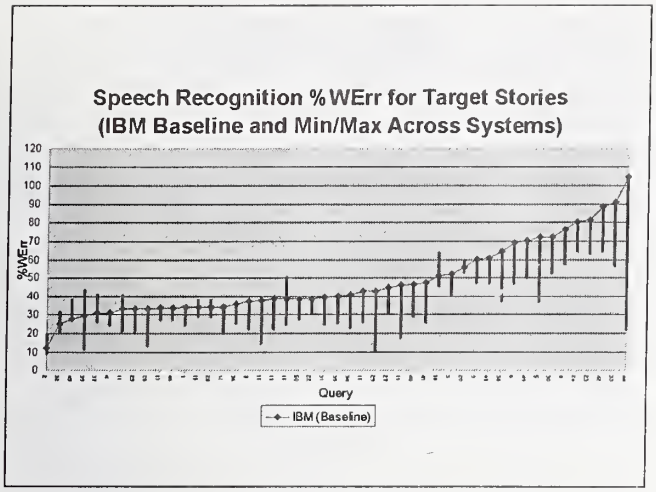


Figure 4. Sorted Target Story Baseline Recognizer Word Error Rate with Min/Max for other Recognizers.

The Word Error Rate for each of the 47 target stories was sorted by increasing error for the Baseline recognizer and plotted along with the min and max Word Error Rates for the other submitted recognizers in Figure 4. Note that the plot for the Baseline recognizer shows a fairly good distribution of error rates across the target story subset.

The recognizer Word Error Rates were determined using procedures and software (sclite) similar to those used in the NIST/DARPA 1996 Hub-4 Broadcast News Continuous Speech Recognition Benchmark Tests. Once scored, the error rates were tabulated by story rather than speaker, segment, or focus condition as in Hub-4.[5] However, the SDR reference transcriptions were not checked and corrected as in Hub-4 and the 1996 Hub-4 orthographic mapping file for lexical normalization was employed which provided only minimal coverage of the SDR test set.[4] Therefore, these SDR Word Error Rates **cannot** be directly compared to those for Hub-4 systems. However, they do provide a point of reference for measuring the relative difficulty of retrieval of stories with respect to recognition accuracy.

4.2 Retrieval Results

Test participants were required to submit a relevance-rank-ordered list of the ID's of the top 1000 stories they retrieved for each topic. But, since the SDR Track employed a known-item task, the results of the retrieval for a topic were considered to be correct only if the target document for the topic appeared at rank 1.

In evaluating retrieval performance, we investigated the measures used in the TREC-5 Confusion Track[2]:

Mean Rank When Found – mean rank at which the target story was found averaged across all topics that retrieved the target story in the top 1000 documents.

Mean Reciprocal Rank – mean of the reciprocal of the rank at which the target story was found over all the topics using 0 as the reciprocal for topics that did not retrieve the story.

Another measure, a plot of the number of topics that retrieve the target document by a certain rank was suggested by ETH.

These measures as well as the rank at which each topic was found were reported in the SDR Appendix of the TREC-6 Notebook.[6]

The results for all three test conditions (Reference, Baseline, and Full SDR) were surprisingly good for an initial evaluation of retrieval of spoken language transcripts. Retrieval rates were very high for the human-transcribed Reference data and most sites showed only small degradation in performance for Full SDR using their own recognition technology. There was generally more degradation using the Baseline recognizer transcripts due to its higher error rate and probably also due to a higher number of "out of vocabulary" (OOV) words. An exception was the Dublin system which showed slightly better performance for the Baseline than the Reference.

Since the results were very good, we decided to employ an additional measure, **Percent Retrieved at Rank 1** across systems and test conditions, which is shown in Figure 5.

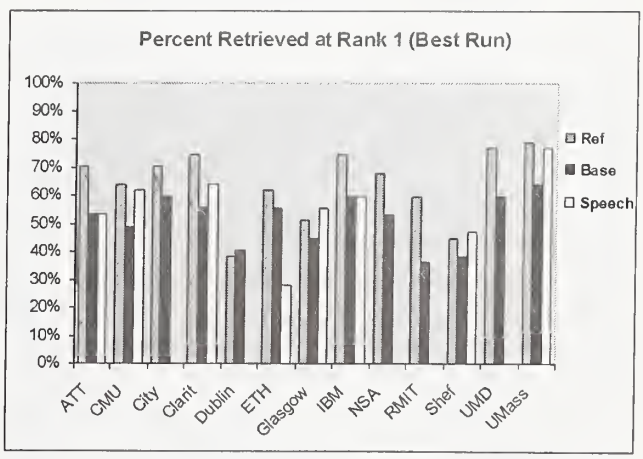


Figure 5. Retrieval rate at rank 1 for all systems and modes (best run).

For Percent Retrieved at Rank 1, the best performance for all three test conditions was achieved by the University of

Massachusetts System (with Dragon Systems Recognition for Full SDR). The UMass system yielded a retrieval rate of 78.7% for the Reference mode, 63.8% for the Baseline mode, and 76.6% for Full SDR. Note that the UMass Reference and Full SDR results differed by only one topic.

A comparable graph for the Mean Reciprocal Rank is given in Figure 6.

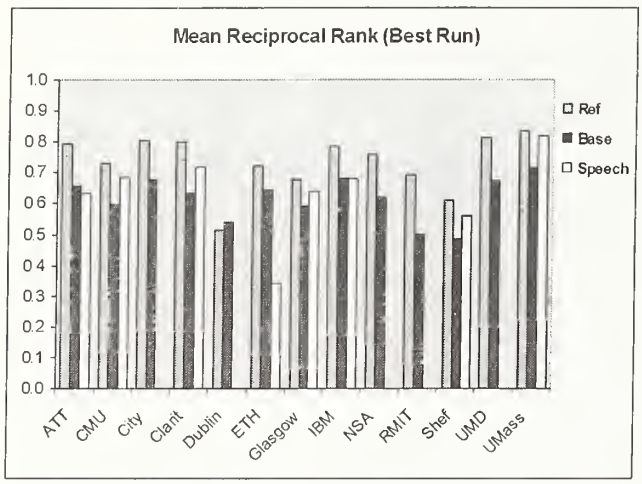


Figure 6. Mean Reciprocal Rank for all systems and modes (best run).

For this evaluation, the Percent Retrieved at Rank 1 and Mean Reciprocal Rank metrics did not show significantly different relative system ranks or trends. It is interesting to note, however, that for the Percent Retrieved at Rank 1 measure only, the Glasgow and Sheffield systems performed more poorly on the Reference condition than on Full SDR most likely due to a bug in processing the Reference transcripts.

Although there is disagreement between the two measures above (Percent Retrieved at Rank 1 and Mean Reciprocal Rank) for the relative ranking of the performance of the retrieval modes for Sheffield and Glasgow, a regression test of the Percent Retrieved at Rank 1 versus the Mean Reciprocal Rank (fig. 7) shows that the two measures were not significantly different for this evaluation.

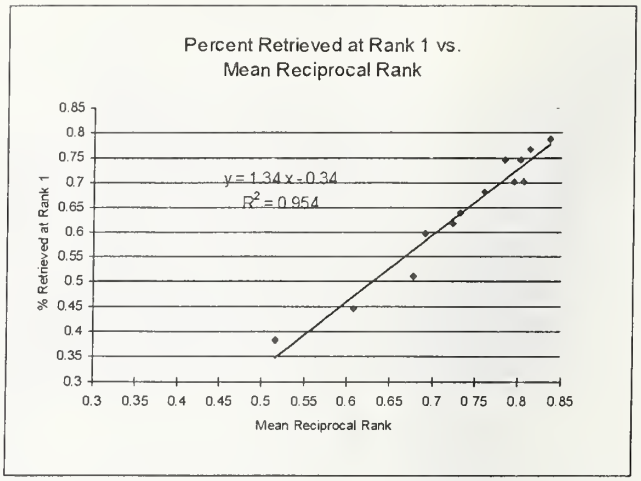


Figure 7. Regression Plot of Percent Retrieved at Rank 1 vs. Mean Reciprocal Rank.

An examination of Percent Retrieved at Rank 1 averaged across systems for each of the topic subset (fig. 8) shows that The "Easy to Recognize" (F0) topic/story set yielded the best performance for all 3 evaluation modes (Ref, Base, and Full SDR) and the "Difficult to Recognize" (FX) topic/story set yielded significantly degraded performance. However, the "Difficult Query" subset yielded even greater performance degradation.

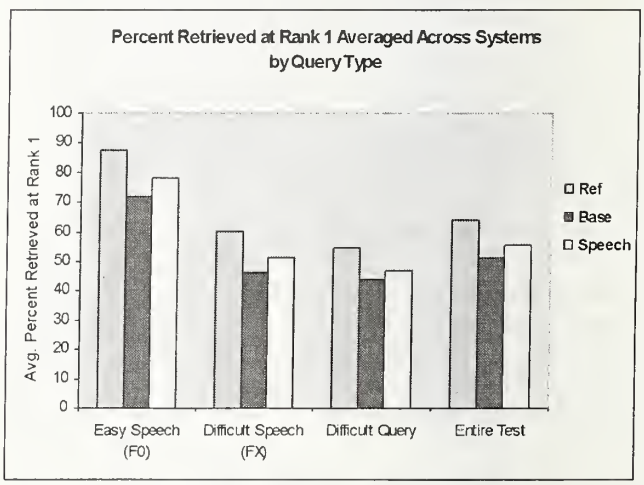


Figure 8. Percent Retrieved at Rank 1 averaged across systems by topic subset.

It is interesting to note that in general, the systems had difficulty with the "Difficult Speech" topics for the Reference retrieval mode (in which the target story texts were not degraded by recognition errors) as well as the Baseline and Full SDR modes (which contained recognition errors.) This may indicate a relationship between language characteristics that degrade recognition and factors that make it difficult to retrieve a spoken document. However, this hypothesis is confounded with

the effect topic difficulty had on retrieval. Figure 8 also shows that the topic difficulty had a much greater effect on retrieval performance than retrieval mode (Reference, Baseline, Full SDR. So, it is clear for future SDR evaluations that if the relationship between recognition and retrieval is to be explored, topic difficulty factors will need to be controlled or at least measured.

In order to look at recognition-related retrieval error trends, we overlayed the sorted Baseline recognizer story Word Error Rate from Figure 4 over the rank at which each story was retrieved (mean, min, and max) across systems for each retrieval mode. This is shown in Figure 9 for the Baseline retrieval condition and in Figure 10 for the Full SDR retrieval condition.

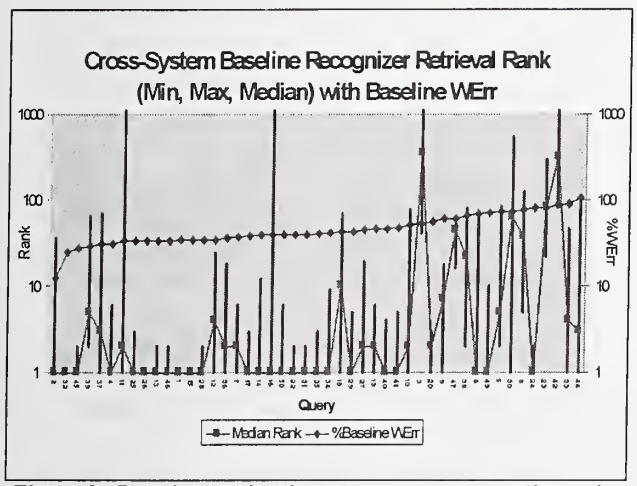


Figure 9. Baseline retrieval mode target story median, min, and max retrieval rank averaged across systems sorted by Baseline Recognizer story Word Error.

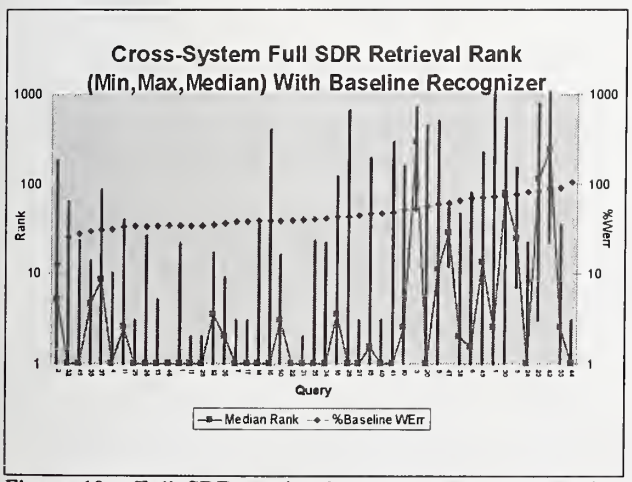


Figure 10. Full SDR retrieval mode target story median, min, and max retrieval rank averaged across systems sorted by Baseline Recognizer story Word Error.

Note that the mean ranks appear to indicate a trend toward increasing retrieval error as the target story recognition error rate increases. However, the same plot for the Reference retrieval condition shown in Figure 11 (which did not suffer from recognition errors) shows a surprisingly similar trend. It appears that difficult-to-recognize stories are also difficult to retrieve -- even if the "perfect" transcribed version of the stories is used for retrieval. This may indicate that there is an indirect relationship between recognition difficulty and retrieval difficulty at the lexical level. One hypothesis is that the complexity of the language itself in these difficult stories is greater. They may also contain fewer key content-bearing words.

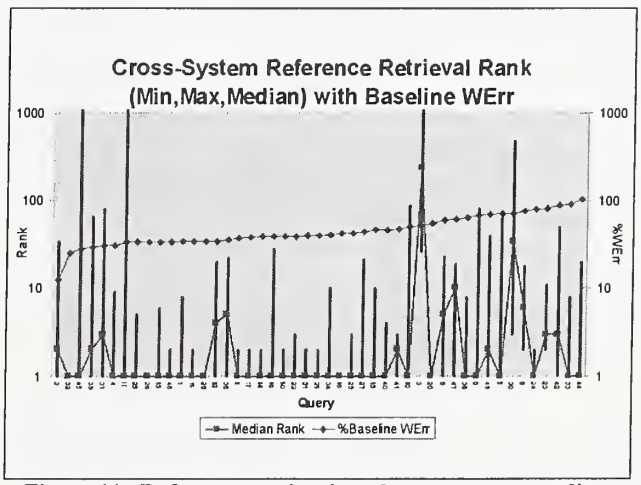


Figure 11. Reference retrieval mode target story median, min, and max retrieval rank averaged across systems sorted by Baseline Recognizer story Word Error.

An interesting exception is found in the results for SDR18. The SDR18 topic was: *Has D.N.A. evidence been used in the Unabomber case?* All 13 systems were able to correctly retrieve the target story for this topic in the Reference Retrieval condition. But, only two of the 13 systems were able to correctly retrieve the story in the Baseline Retrieval condition and only three of the eight systems implementing Full SDR were able to retrieve the document. Upon examining the recognized transcriptions for this story, we find that the key content word, "Unabomber" is never correctly recognized and most systems also had difficulty with a secondary key word, "evidence." These words were most likely "out of vocabulary" for the recognition systems. The retrieval systems failed to retrieve the story since these key content-bearing words were lost. This kind of problem had only a small impact on retrieval using this very small test collection. However, the OOV problem could have a

much more substantial impact on retrieval using realistically large spoken document collections.

Next year, measures of *content word* story recognition may be employed which would provide a better picture of the relationship between recognition accuracy and retrieval performance.

4.3 Statistical Analyses

ANOVA statistical significance tests were also implemented to measure the relative importance of each of the SDR component technologies (recognition and retrieval) in contributing to SDR retrieval performance. When comparing variance from differences in sites and retrieval modes (Reference Baseline, Full SDR), we found that the experimental design was adequate to highlight differences across systems in the Reference and Baseline modes, but not for Full SDR. When comparing variance for the site and the Reference and Baseline retrieval modes, ANOVA evaluation showed that 66.5% of the variance was attributable to the site, 26.3% to the retrieval mode, and only 7.2% was unexplained. (Similar results were also observed if only the subset of sites who performed Full SDR were evaluated.) Note that this result indicates that the retrieval method used was almost three times more important than the transcript (or recognizer) used in differentiating systems.

However, when the variance for the site and Reference, Baseline, and Full SDR retrieval modes was compared, ANOVA evaluation showed that 57.1% of the variance was attributable to the site, 18.9% to the retrieval mode, and **24.0%** to unexplained factors -- thus indicating that effects from the interaction of CSR and IR components were confounded in the results.

This problem would be eliminated only if all IR components were combined with all CSR components and evaluated. An exhaustive cross-component comparison is impractical, at least for the near future. But, CSR sites who produce one-best recognized transcripts will be encouraged to share these with the other participants so that retrieval runs (which are relatively inexpensive) can be run with different recognizer transcripts. This should significantly reduce the problem with unexplained variance.

5. CONCLUSIONS

The first evaluation of SDR technology showed that relatively good known-item retrieval could be achieved for a small collection of broadcast news spoken documents. It also showed that existing speech recognition and information retrieval technologies could be effectively pipelined to perform the task and that spoken document retrieval as well as the underlying component technologies could be evaluated. The initial task, although small by IR standards, brought the IR and CSR communities together and initiated dialogue and collaboration.

During discussions at TREC-6, the SDR participants generally agreed that the test collection would have to be enlarged by at least an order of magnitude before any "real" performance issues would surface. It was also agreed that the known-item task provided insufficient evaluation granularity and should be replaced with an ad-hoc-style relevance evaluation using pooled topics.

Since the test collection this year was far too small to simulate a real deployment of SDR technology, it is impossible to make sweeping judgements about the performance of SDR for real tasks or how well current approaches will scale. It is also, therefore, difficult to make conclusions regarding the relative importance of speech recognition and retrieval accuracy in overall retrieval performance or in the scalability of the technology.

For this test, it appeared that recognition accuracy was not nearly as an important factor as search performance in determining overall retrieval performance. However, it is highly possible that this relationship will not hold for realistically large spoken document collections. In any case, future SDR evaluations with much larger spoken document collections and relevance assessment should help to answer these questions.

6. FUTURE

To progress toward these goals, it is planned that the TREC-7 SDR Track will expand to include a .100-hour test collection and 25 Ad-Hoc-style topics to be developed by the NIST TREC assessors. Like in the TREC Ad-Hoc Track, The retrieved list of stories provided by the participating systems for each topic will be pooled and assessed for relevance by the assessors. Traditional Precision/Recall scoring will then be applied to the results.

For TREC-8, it is planned that the Broadcast News portion of the new TDT-2 Corpus will be used to provide a much larger and more realistic test collection – an order of magnitude larger than the TREC-7 SDR test collection. Current plans call for 1,000 hours/40,000 stories of Broadcast News by the end of 1998.

7. ACKNOWLEDGMENTS

The authors would like to thank David Pallett of NIST for his contributions in development of the concept and design of the SDR Track. We'd also like to thank Jon Fiscus, Darrin Dimmick, and Carol Barnes at NIST for their assistance in tabulating the speech recognition and retrieval scores. Finally, we'd like to particularly thank Salim Roukos and Satya Dharanipragada of IBM for their contribution in providing the baseline recognizer transcriptions for the track.

NOTICE

REFERENCES

[1] Graff, D., Wu, Z., MacIntyre, R., and Liberman, M., *The 1996 Broadcast News Speech and Language-Model Corpus*, Proc. DARPA Speech Recognition Workshop, February 1997.

[2] Kantor, P. and Voorhees, E.M., *The TREC-5 Confusion Track*, Proc. TREC-5, November 1996.

[3] Voorhees, E., Garofolo, J., and Sparck Jones, K., *The TREC-6 Spoken Document Retrieval Track*, Proc. DARPA Speech Recognition Workshop, February 1997.

[4] Garofolo, J., Fiscus, J., and Fisher, W., *Design and preparation of the 1996 Hub-4 Broadcast News Benchmark Test Corpora*, Proc. DARPA Speech Recognition Workshop, February 1997.

[5] Pallett, D., Fiscus, J., and Przybocki, M., *1996 Preliminary Broadcast News Benchmark Tests*, Proc. DARPA Speech Recognition Workshop, February 1997.

[6] Voorhees, E., Garofolo, J., and Sparck Jones, K., *The TREC-6 Spoken Document Retrieval Track*, TREC-6 Notebook, November 1997.

Overview of TREC-6 Very Large Collection Track

David Hawking and Paul Thistlewaite
Co-operative Research Centre For Advanced Computational Systems
Department Of Computer Science
Australian National University
{dave,pbt}@cs.anu.edu.au *

January 10, 1998

Abstract

The emergence of real world applications for text collections orders of magnitude larger than the TREC collection has motivated the introduction of a Very Large Collection track within the TREC framework. The 20 gigabyte data set developed for the track is characterised, track objectives and guidelines are summarised and the measures employed are described. The contribution of the organizations which made data available is gratefully acknowledged and an overview is given of the track participants, the methods used and the results obtained. Alternative options for the future of the track are discussed.

1 Background and Motivation

In the overview of the proceedings of TREC-1, Harman [1992] referred to small early test collections such as Cranfield, CACM and NPL and argued the need for a realistically-sized test collection to facilitate the transfer of laboratory-developed retrieval systems into the field. The 2-gigabyte collection used in TREC-1 was two orders of magnitude larger than previous collections, and legitimately given the label of a *very large test collection.* Indeed, given the state of contemporary hardware and indexing software, it posed considerable challenges to participants.

Two gigabytes remains a realistically-sized test for text retrieval applications typical of universities, research organisations, newspapers, businesses and government departments. However, it is clear that some organisations such as patent offices and future digital libraries will demand retrieval services over collections at least two orders of magnitude larger, despite trends toward distributed information retrieval. There are already collections of the 100 gigabyte scale in the commercial world and Web search engines such as HotBot claim to index in excess of 50 gigabytes.

Accordingly, in line with the initial TREC charter of realism, a need was identified for a test collection significantly larger than that used in mainstream TREC. It was not intended to replace the main TREC collection but rather to be used in a special-interest Very Large Collection (VLC) track to allow interested developers of commercial and research retrieval systems to investigate the scalability of their methods. It would also help to verify that such systems did not suddenly cease to operate due to machine or operating system limits on virtual addressing, file system

*The authors wish to acknowledge that this work was carried out within the Cooperative Research Centre for Advanced Computational Systems established under the Australian Government's Cooperative Research Centres Program.

size etc. and allow some effectiveness comparison of systems currently operating with very large collections. The proposed collection size was 20 gigabytes, a factor of ten larger than that used in the TREC mainstream task. This collection size seemed feasible, as hardware and software improvements since TREC-1 had dramatically reduced the difficulty of working with gigabyte-scale collections.

The value of a test collection lies not only in the data itself but in the availability of judgments of its documents as to relevance to a large set of research topics. Complete sets of judgments are available for some test collections but are not affordable for TREC-sized collections. (At an optimistic judging rate of 500 documents per judge per working day, complete judgment of a collection of one million documents requires about eight person-years *per topic*!) TREC approximates a complete set of judgments for its topics by manually judging only those documents in the pool retrieved by a [hopefully] diverse set of automatic retrieval systems and deeming that un-judged documents are irrelevant. This allows recall-oriented measures to be determined with a reasonable (but not perfect) degree of confidence.

Assessment resources available to the VLC track are not sufficient to support recall-oriented measures over a 20-gigabyte collection. Even if sufficient resources were available to support the TREC pooling method, that method is not likely to be effective in the VLC context. For any given topic there may be ten times as many relevant documents as in the standard TREC task yet the reduced number of participating systems is likely to mean that fewer are judged.

Accordingly, effectiveness measures in the VLC track were confined to the precision dimension. It was envisaged that TREC participants could demonstrate the speed and effectiveness merits of their system on the main AdHoc task and then, if interested in larger collections, demonstrate how speed was affected by a ten-fold increase in data size and (hopefully) confirm that speed results were not achieved at the expense of lost precision.

A trial run of the VLC track took place in TREC-5 (1996) using CDs 1-4 of the TREC set (a total of 4.28 gigabytes). Four groups submitted runs, judgments made by Canberra assessors were validated against those in Washington and various issues were clarified for the running of the track proper at TREC-6.

2 The Organisers

The VLC track (like the pre-track in TREC-5) has been organised by the Advanced Computational Systems Cooperative Research Centre (ACSys), whose core participants are the Australian National University, the Commonwealth Scientific and Industrial Research Organisation, Fujitsu, Sun and DEC. Support for the VLC track is a natural extension of ACSys research interests in scalable computing and large datasets.

With full support from NIST and the TREC program committee, ACSys collected the additional data to make up the VLC and supplied the human, financial and machine resources to format and distribute the data. It also recruited and employed the VLC assessors.

3 The Participants

Fourteen groups, including 6 universities, received VLC data tapes. One registered very late and was unable to read the tapes. In the end, seven groups submitted runs, comprising four universities and three commercial groups: ANU/ACSys, City, UMass, UWaterloo, AT&T and IBM (two separate groups).

4 The Data

Additional information on the Very Large Collection is available on the VLC web page [Hawking et al. 1997].

A 20.14 gigabyte collection (including all five TREC CD-ROMs) was assembled with assistance from a large number of data holders. From it, a uniform 10% sample was defined for use as a baseline.

The additional (non-NIST) data was distributed on DAT (DDS-1) format tapes due to logistical and economic difficulties with using CD-ROMs. Participants reported some difficulties in reading these tapes but only in one case (a late starter) were these responsible for a non-submission. The final set of tapes was shipped to all registered participants (at the time) on June 20, 1997, allowing participants roughly nine weeks to work on the task up to the submission deadline of September 8.

4.1 Access to the VLC Data

Access to the data (except for USENET news data) is subject to the terms and conditions of the TREC data permission forms. Copyright owners only granted permission to distribute the data on this basis. These owners are listed in the Acknowlegements below. Permissions were obtained from controllers of all websites used as sources of documents.

4.2 Overview of Data

The VLC data is somewhat biased by the inclusion of roughly 8.7 gigabytes of USENET news postings to make up the target 20 gigabytes. This data has a significantly different character to the data on CDs 1-5. However, the remainder of the non-NIST data in the VLC adheres reasonably well to the earlier TREC pattern and represents a diversity of sources covering government agencies (eg. Australian Department of Industrial Relations), parliamentary proceedings (Canadian and Australian Hansards) and newspapers (eg. Glasgow Herald and Financial Times). For the first time, HTML documents downloaded from the Internet are included (eg. CSIRO and Australian university websites). Also for the first time, there is a large quantity of legal data including both laws and judgments, thanks to the Australian Attorney General's Department. The latter is mostly in HTML format.

Collections in the new VLC data are typically larger than those on CDs 1-5. However, addition of the new data has not altered the minimum or maximum document length figures. Average document length has declined slightly, from 3.2 kilobyte for CDs 1-5 to 2.8 kilobyte for the entire VLC.

The 10% baseline sample was created by selecting every 10th compressed file and then manually removing an arbitrary handful of files to bring the sample to a closer approximation of 10%. Average and minimum document lengths changed by negligible amounts but the longest document in the baseline dropped to 2.8 MB from 6.2 MB.

4.3 International Balance

The international balance of the data is significantly different to the combined NIST data, of which 90% is sourced in the U.S. Ignoring the NEWS and Project Gutenberg collections, whose origins are mixed but U.S.-dominated, the remaining 11.3 gigabytes is sourced roughly 41% from the U.S., 44% from Australia, 10% from England, 4% from Scotland and less than 1%

Table 1: Crude breakdown of VLC, VLC assessment pool and VLC relevant set by source.

Source	# Documents	# Documents judged	# Relevant documents
TREC6 docs.	556,077(7.4%)	1608(18.9%)	631(21.7%)
Other NIST docs.	1,078,166(14.4%)	3426(40.3%)	1202(41.3%)
All ACSys-collected docs.	5,857,805(78.2%)	3477(40.9%)	1076(37.0%)
- USENET news docs.	4,400,657(58.7%)	2001(23.5%)	552(19.0%)
- ACSys non-USENET docs.	1,457,148(19.4%)	1476(17.3%)	524(18.0%)

Table 2: Probability of retrieval and probability of relevance for documents from different sources. (Obtained by dividing the raw frequencies in table 1 by the number of documents from each source.) The last column gives the probability for each source that a document in the assessment pool is actually relevant.

Source	Pr(retrieved)	Pr(relevant)	Pr(relevant|retrieved)
TREC6 docs.	0.00289	0.00113	0.392
Other NIST docs.	0.00318	0.00111	0.351
All ACSys-collected docs.	0.000593	0.000184	0.309
- USENET news docs.	0.000455	0.000125	0.276
- ACSys non-USENET docs.	0.00101	0.000360	0.355

Table 3: Contributions of individual ACSys collections to the VLC pool and the VLC relevant set. The probability ratios are computed by calculating the probability that a document from this source will be part of the pool (or part of the relevant set) and dividing this by the corresponding probability for all NIST-collected documents.

	Collection		Pool			Relevant Set		
Source	MB	# docs	# docs	% of pool	Prob. Ratio	# docs	% of rel. set.	Prob. Ratio
AAG	1874.5	61,566	230	2.7%	0.133	59	2.0%	0.094
ADIR	775.0	42,841	9	0.1%	0.068	1	0.0%	0.021
APLT	1539.8	421,681	501	5.9%	0.386	185	6.4%	0.391
AUNI	724.8	81,334	134	1.5%	0.535	40	1.4%	0.438
FT	526.7	202,433	259	3.0%	0.415	100	3.4%	0.440
GH	393.6	135,477	251	2.9%	0.601	107	3.7%	0.704
NEWS01	954.5	446,106	180	2.1%	0.131	65	2.2%	0.130
NEWS02	943.1	450,027	221	2.6%	0.159	63	2.2%	0.125
NEWS03	936.6	482,395	228	2.7%	0.153	56	1.9%	0.104
NEWS04	966.0	83,145	233	2.7%	0.157	61	2.1%	0.113
NEWS05	1169.7	590,202	325	3.8%	0.179	91	3.1%	0.137
NEWS06	1120.6	571,891	260	3.1%	0.148	47	1.6%	0.073
NEWS07	1080.1	520,282	240	2.8%	0.150	60	2.1%	0.103
NEWS08	1727.9	856,609	314	3.7%	0.119	109	3.7%	0.113
PGUT	430.3	3,303	30	0.4%	2.949	5	0.2%	1.350
WEB01	141.9	8,513	62	0.7%	2.364	27	0.9%	2.828

from Canada. These proportions reflect the availability of data rather than any goal of the organisers. The proportion of non-English-language text in the VLC is negligible.

4.4 Formatting

A variety of `flex` programs and `perl` scripts were used to convert supplied data into VLC format. The `wget` program was used to download web pages from the web sites for which permission to distribute was granted. Some effort was made to eliminate encoded binary data from within news items but one VLC participant has indicated that this was not totally successful. Efforts were also made to eliminate web pages which explicitly claimed copyright for an organisation other than the host site.

Data within the `tar` files on the VLC tapes was formatted in the same way as the data on the CD-ROMS - as a directory hierarchy of multi-document files compressed using the standard Unix `compress` utility. Document identifiers were structured to allow unambiguous identification of collection, sub-directory and filename. Every document contained the four essential "SGML" markers delimiting documents and document identifiers. A program `coll_check` was used to check that each document conformed to this elementary structure and that document identifiers were unique. No effort was made to ensure that resulting documents conformed to SGML standards.

5 The Task

Full guidelines for the VLC track are available on the VLC web page [Hawking et al. 1997]. In essence, participants were required to process queries generated from the TREC-6 AdHoc topics (301-350) over both the baseline and the VLC datasets and to return for assessment only the first 20 documents retrieved in each case. Elapsed times (as would have been observed by a human with a stopwatch) for indexing the datasets and processing queries were recorded and system details and costs as well as disk space requirements were reported via a questionnaire. The focus was on the ratios of the various measures (see below) for the VLC run compared with the baseline run.

All retrieved documents were judged. Only one baseline and one VLC run were permitted due to assessment resource limitations.

Participants were given the choice of comparing the measures for FIXED QUERIES derived either manually, interactively (e.g. over CD4 and CD5 in the AdHoc task) or automatically OR for queries which were expanded automatically over the dataset in use. No interaction with queries was permitted using either the baseline or the VLC collections.

6 The Measures

M1. Completion. (Can the system process data of this size at all?)

M2. Precision@20

M3. Query response time (Elapsed time as seen by the user)

M4. Data Structure Building time (Elapsed time as seen by the user)

M5. Gigabyte-queries/hour/kilodollar. (Modified to incorporate the size of the data set.)

M4 represented the minimum possible elapsed time from receiving the data until the data structures necessary to process the queries used in M3 were built, using the chosen hardware and indexing software. Time to actually read the CD-ROMs and DATs was excluded. The starting point was the compressed data files on disk after copying the CD-ROMs and unpacking the DAT tarfiles. M4 included the time to build all structures (such as inverted files) which are necessary to process the final query. Groups building phrase dictionaries, thesauri, co-occurrence matrices etc. for use in query building (NOT in query processing) were encouraged to report these times separately as M4R.

7 The Assessments

Three judges were employed to assess the VLC document pool. One was a PhD student and former research assistant in Asian Studies, another was a research assistant in Sociology and the other a recent Honours graduate in Economic History. The first judge was also employed in the TREC-5 pre-track. Some overlap between judges was organised as a sanity check and no significant discrepancies were found.

The document pool (derived from both baseline and VLC submissions) contained 8511 documents of which 2909 documents were judged relevant.

Of the total VLC pool, 1465 documents (17%) were also judged (against the same topic) by the NIST assessors as part of the AdHoc pool. NIST and ACSys judges agreed on 83% of cases.

8 Makeup of VLC Judgment Pool and Relevant Set

It would have been unfortunate had all of the documents in the VLC judging pool (or the VLC relevant set) come from CDs 4 & 5 or indeed from only the NIST-collected documents. Table 1 shows that this was not the case. As might be expected, given that the topics were not oriented toward the VLC data, the probability of a given document being selected by a retrieval system was significantly lower for the ACSys-collected documents than for the NIST-collected ones. Table 2 shows that USENET news documents were 6.7 times less likely to be retrieved than NIST-collected ones. The corresponding figure for ACSys-collected non-USENET documents was 3.0.

The probability that a document in the judging pool was relevant did not differ much between the NIST-collected and ACSys-collected, non-USENET documents. However, a USENET document in the pool was only 76% as likely to be judged relevant as other documents in the pool.

Table 3 shows the breakdown of ACSys-collected documents by individual collection. Perhaps surprisingly given the nature of some of the collections, each collection contributed at least one document to the relevant set.

8.1 Was the Baseline Collection an Unbiased Sample?

This is an important question, because it may determine the "scalability" of early precision and perhaps influence other measures.

The process of selecting the baseline subset has been described above. The baseline subset contains 10.02% of the VLC data and 10.05% of the documents.

Of the 4833 different documents retrieved in the runs over the full VLC 460 (9.52%) were actually baseline documents. The proportion of documents in the VLC and the sample which were retrieved by VLC (not baseline) runs were 0.0006451 and 0.0006108. A test of one-sample

Table 4: Groups completing the VLC task. All groups attempted the full 20 gigabyte task but, due to problems, IBMg(Brown) actually used only 17.8 gigabytes.

Group	Query Gen.	Terms/Query	Stems	Query Opt.	Baseline Hardware	VLC Hardware
ANU	Auto_long	30	Yes	Yes	1 x DEC Alpha	8 x DEC Alpha
ATT	Auto_long	27	Yes	No	1 x SGI R10000	5 x SGI R10000
City	Auto_long	25	Yes	No	1 x Sun Ultra	1 x Sun Ultra
IBMs(Franz)	Auto_short	13 + Expand	Morphing	No	1 x IBM RS/6000	1 x IBM RS/6000
IBMg(Brown)	Auto_short	20	Morphing	No	1 x IBM RS/6000	6 x IBM RS/6000
UMass	Auto (title + desc)	66	Yes	No	1 x Sun Ultra	1 x Sun Ultra
U Waterloo	Manual	5.5	No	No	4 x Cyrix PC	4 x Cyrix PC

Table 5: M2: Precision at 20 documents retrieved. The asterisked items for IBMg(Brown) may have been higher if the full data had been used.

Group	Baseline	VLC	Ratio
City	0.320	0.515	1.61
ATT	0.348	0.530	1.52
ANU	0.356	0.509	1.43
UMass	0.387	0.505	1.31
IBMg(Brown)	0.275	0.361*	1.31*
U Waterloo	0.498	0.643	1.29
IBMs(Franz)	0.271	0.348	1.28

Table 6: M3: Average Query Processing Time (Elapsed minutes per 50 queries.) Figures in parentheses for IBMg(Brown) are scaled up by 20.1/17.8 to compensate for the smaller data size used. The baseline figure for the starred IBMs(Franz) run was derived by linear scaling of the VLC run.

Group	Baseline	VLC	Ratio
IBMg(Brown)	16.5	47.2(53.3)	2.86(3.23)
ANU	10.1	42.1	4.17
ATT	0.45	1.93	4.30
U Waterloo	0.189	1.12	5.93
City	6.3	61.4	9.75
IBMs(Franz)	886*	8857	10.0*
UMass	34.6	346	10.0

Table 7: M4: Data Structure Building Time (Elapsed Hours). Figures in parentheses for IBMg(Brown) are scaled up by 20.1/17.8 to compensate for the smaller data size used. The baseline figure for the starred IBMs(Franz) run was derived by linear scaling of the VLC run.

Group	Baseline	VLC	Ratio
ATT	0.768	2.57	3.34
IBMg(Brown)	3.23	28.4(32.1)	8.79(9.93)
IBMs(Franz)	86.9*	869	10.0*
UMass	6.85	69.14	10.1
City	9.9	103	10.4
U Waterloo	0.42	4.48	10.7
ANU	1.41	15.6	11.1

Table 8: MS: Data Structure Sizes (gigabytes). Figures in parentheses for IBMg(Brown) are scaled up by 20.1/17.8 to compensate for the smaller data size used. The baseline figure for the starred IBMs(Franz) run was derived by linear scaling of the VLC run. (Waterloo indicated at the conference that the sizes given in their questionnaire response and reported here may be higher than the correct values. Revised values are not yet available.)

Group	Baseline	VLC	Ratio
U Waterloo	3.36	30.9	9.20
City	2.47	23.6	9.55
ANU	0.626	6.06	9.68
IBMs(Franz)	1.21*	12.1	10.0*
IBMg(Brown)	1.21	10.8(12.2)	8.93(10.1)
ATT	1.23	13.02	10.6
UMass	1.22	11.43	?

Table 9: M5: Gigabyte-queries per hour per kilodollar.

	Baseline			VLC		
Group	Queries/Hr	kilo$	gB-Q/Hr/kilo$	Queries/Hr	kilo$	gB-Q/Hr/kilo$
U Waterloo	15873	7.44	4267.0	2678	7.44	7198.0
UMass	4392	45.7	3.8	439	45.7	3.8
ATT	6667	115	116	1554	394	78.9
City	476	14.2	67.0	48.9	14.2	68.8
ANU	297	23.9	24.8	71.3	95.1	15.0
IBMg(Brown)	182	17.3	21.0	63.6	123	10.3
IBMs(Franz)	3.39	30	0.226	0.339	30	0.226

proportion (with finite sample correction) shows that the sample proportion lies within the 95% confidence interval. Hence, there is no reason to conclude that the sample is biased with respect to proportion of retrieved documents.

9 Characteristics of Submitted Runs

The seven groups which passed the finishing post are listed in Table 4, which gives salient features of the methods used.

9.1 Hardware Used

A large range of hardware platforms were used, ranging from single workstations through clusters of PCs to large scale SMP systems. IBM, DEC, Sun, SGI and Cyrix hardware was used.

City used a single Sun workstation. UMass and ATT used a part of shared-memory multiprocessor (SMP) systems. ANU, IBMs(Franz), IBMg(Brown) and Waterloo used networks or clusters of workstations (COWs).

Attempts to calculate "bang per buck" measures are not especially meaningful because:

1. Groups used hardware they had access to rather than explicitly choosing it for the task. Their systems may have run just as fast on much cheaper hardware.

2. Few groups were able to run their system in dedicated mode. It is difficult to control for the effect of other users.

3. It is difficult to derive a comparable dollar value for a group which used a fraction of a very expensive system.

9.2 Approaches Taken

IBMg(Brown), ANU and ATT attempted to reduce the growth in query-processing time due to increased data size by adding more hardware. IBMs(Franz) actually did something similar but added all the individual times to produce a single-system time.

IBMg(Brown) used a collection fusion approach with no attempt to normalise rankings between the six parts of the collection. ATT divided the collection into 5 separately indexed pieces. Once indexes were built there was an exchange of document frequencies until all processors held correct global *df*s. ANU divided the collection and communicated *df* information (if necessary) at query processing time.

Waterloo used the same cluster of four PCs in both baseline and VLC runs. Waterloo also divided the collection into pieces but, due to use of distance-based relevance scoring, there was no resulting difference in results.

UMass and City essentially processed the VLC using a single processor although in the former case, the processor was one of four in an SMP system.

IBMs(Franz) was the only group not to run queries sequentially.

9.3 Query Generation

All query processing times reported were for the processing of fixed queries ie. did not include automatic feedback. City used automatic feedback over the collections but the query expansion time was not included in the tabulated figures.

Waterloo were the only group to use manually generated queries. These were the result of refinement by interaction with CD4/CD5 and other non-VLC documents. Other groups used automatic queries generated from all or various parts of the topic statement.

10 The Results

1. The shortest queries (5.5 terms, Waterloo) led to both the fastest processing and the best early precision. (Tables 4, 5 and 6) These queries were manually generated.

2. All runs showed at least 28% improvement in early precision for the VLC over the baseline. (Table 5)

3. Query processing time increased linearly with data size for uni-processor systems. Query processing time did not increase linearly for the Waterloo submissions which used the same hardware for both runs. (Table 6) It is understood that this is because a constant-time component of their algorithm ceased to be negligible when the data-size dependent component became very small, as was the case for their baseline run.

4. It is possible to reduce the query processing time scaling factor by scaling the hardware, but this year no group achieved a scaling factor of anything close to unity. (Table 6)

5. Data structure building is normally considered to be embarassingly parallel provided that the separately indexed pieces are evenly sized and not too small. However, only ATT exploited parallelism to bring the ratio below 10. (Table 7)

6. The fastest indexing rate was 7.84 gigabytes per elapsed hour (ATT) albeit on a very large machine. (Table 7)

7. Data structure sizes tended to increase linearly with the size of the raw data. (Table 8)

8. Data structure sizes for the VLC ranged from 6.06 gigabytes (ANU) to 30.9 gigabytes (Waterloo)[1]

9. Given the difficulties outlined above of assigning comparable dollar values to hardware actually used, it is difficult to place much emphasis on the results presented in table 9.

11 Discussion and Conclusions

The VLC track results clearly demonstrate that there are a number of retrieval systems for which query processing over 20 gigabytes is not at all daunting.

Good performance on the VLC size does not demand the use of exotic and expensive hardware. The best evidence for this conclusion is the Waterloo run over the full 20 gigabyte collection using an etherneted cluster of four commodity PCs whose total cost was only US$7,440. This run (using manually generated queries):

- retrieved an average 12.8 relevant documents in the first 20,
- indexed the data at a rate of 4.5 gigabytes per elapsed hour, and
- processed queries at a rate of 2678 queries per elapsed hour.

[1]This figure may not be correct. See the note in Table 8.

The only apparent downsides to the method used were the amount of disk space required and the use of manual queries.

The increase in early precision with the increase in data size is an interesting effect whose explanation is to be addressed elsewhere. It may be possible for groups to exploit the effect by using quicker, lower-quality algorithms on the VLC compared to the baseline. If judged well, early precision would remain constant but scalability would improve.

The processing of the 20 gigabyte collection should not be seen as an end in itself but rather as a way of predicting how retrieval systems will perform as data sizes grow to the multi-terabyte level. To make such predictions one must consider both the query processing performance of the system at a particular level and its scaling factor (after convincing oneself that the system will continue to scale at that rate). If query processing time grows in proportion to data size, then seconds at the gigabyte level will become hours at the multi-terabyte level. On the other hand, query processing times which remain constant despite data growth are not attractive if they already take hours at the gigabyte level.

12 The Future

12.1 Increasing Collection Size?

It is doubtful that increasing the size of the VLC by a small factor would improve the value of the track. Growth by a further order of magnitude would extend the scope of the problem but could dramatically increase the cost of participating in the track and possibly the cost of organising the track.

Considerable difficulty has been experienced in persuading organisations to make data available, due to concerns about data security or because of the resources required by the data donor to extract the data in a suitable form, or because the data holder itself does not have permission to distribute some of it.

Consequently, the only visible options for large increases in collection size are:

1. adding huge amounts of USENET news archived by the University of Waterloo;
2. approaching the Internet Archive (`http://www.archive.org/`);
3. replicating the existing data.

Participants in the VLC workshop at TREC-6 strongly expressed the view that an effort should be made to build the VLC up to 100 gigabytes for TREC-7, even if all the additional data is USENET news items. Interest was also expressed in addressing the problem of dealing with duplicate or near-duplicate documents.

12.2 Standardising Systems

It has been suggested that an attempt should be made to negate differences in hardware by defining a benchmark whose results could be used to scale timing results on the tasks. Unfortunately, it is likely that this would raise as many questions as it answered, as the algorithms employed differ enormously in the relative demands they place on CPU, memory, disk and network components.

12.3 Possible Revisions to Track Guidelines

It may be possible to allow more than one submission per group in 1998, provided that the size of the assessment pool does not grow too much.

12.4 Goals, Challenges and Purposes

The VLC track serves a number of different purposes:

- It complements mainstream TREC by allowing qualification, measurement and comparison of systems on the efficiency dimension.
- It may stimulate the development of algorithms whose space and time cost grows less rapidly than the increase in data size.
- It encourages consideration of the most suitable hardware and software architectures for tackling huge text collections of the (near) future.

This year failed to produce a set of results showing all of: query processing time over twenty gigabytes < 1.0 sec, precision@20 > 0.5, indexing rates > 10 gigabytes/hr and scaling factors ≈ 1.0. However, there are indications that such a combination may be possible and that achieving it may not require excessively expensive hardware.

Acknowledgements

We are indebted to Mark Sanderson and Jon Ritchie of Glasgow University who arranged for the release of data from the Glasgow Herald and for the additional data from the Financial Times and to Gordon Cormack and Rob Good of the University of Waterloo who supplied huge quantities of archived USENET news. Thanks also to Donna Harman for official support on behalf of NIST and the TREC Program Committee without which the VLC would not have come into being. Donna Harman, Ellen Voorhees and Dawn Tice(Hoffman) of NIST provided advice, support and practical assistance.

We acknowledge the work of Jason Haines, Tim Potter and Nick Craswell in formatting the new data and to Deborah Johnson, Sonya Welykyj and Josh Gordon in assessing submissions.

We would also like to express our appreciation to the organisations who gave permission to use their data in the VLC track. The willingness of organisations to make available commercially valuable data is a vote of confidence in TREC and in the integrity of its participants. VLC donor organisations in 1997 were: Canadian House of Commons (for Canadian Hansard); Australian Department of Defence (for the Defence Home Page); Australian Computer Society (for ACS web pages); National Library of Australia (for NLA web pages); Australian Broadcasting Commission (for Radio National web pages); Commonwealth Scientific and Industrial Research Organisation (for CSIRO web pages); Australian National University (for web pages); Victorian University of Technology (for web pages); Latrobe University (for web pages); Ballarat University (for web pages); Adelaide University (for web pages); Charles Sturt University (for web pages); University of Tasmania (for web pages); Edith Cowan University (for web pages); Murdoch University (for web pages); University of Newcastle, NSW (for web pages); Financial Times, London (for newspaper data 1988-1990); Caledonian Newspapers Ltd, Scottish Media Group (for Glasgow Herald data, 1995-97); Parliament of Australia (for parliamentary data including Hansard 1970-1995); CAUT Clearinghouse in Engineering (for web pages); Australian Attorney-General's Department (for legislation, court decisions and other legal data); Uniserve Coordinating Centre (for web pages); Australian Department of Industrial Relations (for industrial relations data).

Bibliography

HARMAN, D. K. Ed. 1992. *Proc. First Text Retrieval Conference (TREC-1)* (Gaithersburg, MD, November 1992). U.S. National Institute of Standards and Technology. NIST special publication 500-207.

HAWKING, D., THISTLEWAITE, P., AND CRASWELL, N. 1997. *TREC Very Large Collection (VLC) web page.* `http://pastime.anu.edu.au/TAR/vlc.html/`: ACSys Cooperative Research Centre, Australian National University.

Using Clustering and SuperConcepts Within SMART : TREC 6

Chris Buckley*, Mandar Mitra†, Janet Walz*, Claire Cardie†

Abstract

The Smart information retrieval project emphasizes completely automatic approaches to the understanding and retrieval of large quantities of text. We continue our work in TREC 6, performing runs in the routing, ad-hoc, and foreign language environments, including cross-lingual runs. The major focus this year is on trying to maintain the balance of the query – attempting to ensure the various aspects of the original query are appropriately addressed, especially while adding expansion terms. Exactly the same procedure is used for foreign language environments as for English; our tenet is that good information retrieval techniques are more powerful than linguistic knowledge. We also give an interesting cross-lingual run, assuming that French and English are closely enough related so that a query in one language can be run directly on a collection in the other language by just "correcting" the spelling of the query words. This is quite successful for most queries.

Introduction

For over 30 years, the Smart project at Cornell University, under the direction of the late Gerry Salton, has been investigating the analysis, search, and retrieval of heterogeneous text databases, where the vocabulary is allowed to vary widely, and the subject matter is unrestricted. Our belief is that text analysis and retrieval must necessarily be based primarily on a study of the available texts themselves. The community does not understand natural language well enough at the present time to make use of a more complex text analysis. Knowledge bases covering the detailed structure of particular subject areas, together with inference rules designed to derive relationships between the relevant concepts, are very difficult to construct, and have not yet been proven to aid in general retrieval.

Fortunately very large text databases are now available in machine-readable form, and a substantial amount of information is automatically derivable about the occurrence properties of words and expressions in natural-language texts, and about the contexts in which the words are used. This information can help in determining whether a query and a text are semantically homogeneous, that is, whether they cover similar subject areas. When that is the case, the text can be retrieved in response to the query.

Automatic Indexing

In the Smart system, the vector-processing model of retrieval is used to transform both the available information requests as well as the stored documents into vectors of the form:

$$D_i = (w_{i1}, w_{i2}, \ldots, w_{it})$$

where D_i represents a document (or query) text and w_{ik} is the weight of term T_k in document D_i. A weight of zero is used for terms that are absent from a particular document, and positive weights characterize terms actually assigned. The assumption is that t terms in all are available for the representation of the information.

*SabIR Research

†Department of Computer Science, Cornell University, Ithaca, NY 14853-7501

The basic "tf*idf" weighting schemes used within SMART have been discussed many times. For TREC 6 we use the same basic weights and document length normalization as were developed at Cornell by Amit Singhal for TREC 4. Tests on various collections show that this indexing is reasonably collection independent and thus should be valid across a wide range of new collections. No human expertise in the subject matter is required for either the initial collection creation, or the actual query formulation.

The same phrase strategy (and phrases) used in all previous TRECs ([2, 1, 3, 4, 5]) are used for TREC 6. Any pair of adjacent non-stopwords is regarded as a potential phrase. The final list of phrases is composed of those pairs of words occurring in 25 or more documents of the initial TREC 1 document set. Phrases are weighted with the same scheme as single terms.

Text Similarity Computation

When the text of document D_i is represented by a vector of the form $(d_{i1}, d_{i2}, \ldots, d_{it})$ and query Q_j by the vector $(q_{j1}, q_{j2}, \ldots, q_{jt})$, a similarity (S) computation between the two items can conveniently be obtained as the inner product between corresponding weighted term vectors as follows:

$$S(D_i, Q_j) = \sum_{k=1}^{t} (d_{ik} * q_{jk}) \tag{1}$$

Thus, the similarity between two texts (whether query or document) depends on the weights of coinciding terms in the two vectors. The SuperConcept similarity function described later will be a slight variant of this inner product function, but still depends on weights of coinciding terms.

System Description

The Cornell TREC experiments use the SMART Information Retrieval System, Version 13, and most were run on a dedicated Sun Sparc 20/51 with 160 Megabytes of memory and 33 Gigabytes of local disk (some supporting runs were made on a Sun UltraSparc 1/140).

SMART Version 13 is the latest in a long line of experimental information retrieval systems, dating back over 30 years, developed under the guidance of G. Salton. The new version is approximately 44,000 lines of C code and documentation.

SMART Version 13 offers a basic framework for investigations of the vector space and related models of information retrieval. Documents are fully automatically indexed, with each document representation being a weighted vector of concepts, the weight indicating the importance of a concept to that particular document (as described above). The document representatives are stored on disk as an inverted file. Natural language queries undergo the same indexing process. The query representative vector is then compared with the indexed document representatives to arrive at a similarity (equation (1)), and the documents are then fully ranked by similarity.

SMART is highly flexible and very fast, thus providing an ideal platform for information retrieval experimentation. Documents for TREC 5 are indexed at a rate of over a Gigabyte an hour, on hardware costing under $10,000 new. Retrieval speed is similarly fast, with basic simple searches taking much less than a second a query.

Ad-hoc

Ad-hoc Methodology

Automatic query expansion using pseudo relevance feedback has traditionally been very useful in the ad-hoc task. In this approach, a set of documents is initially retrieved in response to a user query; the top-ranked documents are assumed to be relevant (without any intervention from the user); low-ranked documents are optionally assumed to be non-relevant; and these documents are then used in the Rocchio feedback method to expand the query.

There are two steps in the above procedure that we can improve. First, the initial retrieval can be improved so that the assumption of relevance for the top-ranked documents is more accurate. Secondly, the expansion procedure can be improved to yield a better final query. For our TREC 6 ad-hoc runs, we attempt to improve both these steps.

Better initial retrieval. The query coverage (QC) algorithm used in our TREC 5 ad-hoc run is reused for the TREC 6 task to improve the set of documents assumed relevant. Briefly, this algorithm retrieves a few (50, say) documents using the simple vector similarity and computes a new similarity between queries and top-ranked documents based on whether query terms occur close to each other in a document (within a window of 50 terms, say), and whether the matching query terms are "independent" (in the sense that their occurrences are uncorrelated, rather than the occurrence of one term being a reasonable predictor of the presence of another). The top 50 documents are re-ranked based on this new similarity and the top 20 in the resulting ranking are assumed relevant. Using this refined set of 20 documents in the feedback process yielded good improvements at TREC 5 [5].

We also experimented with other techniques to improve the initial retrieval. We used natural language processing techniques to identify phrases in queries and documents, hoping that the high-quality phrases identified this way could be used to better predict the relevance of a document. We found however that phrases have little effect on the ordering of documents at top ranks and thus cannot be used to significantly improve the quality of the initially retrieved set. (The details of our experiments are reported in [6].) We therefore did not use this method in our official submissions.

Another approach involved the use of Boolean filters to refine the initially retrieved set. By filtering out top-ranked documents that fail to satisfy certain Boolean constraints, we can eliminate several non-relevant documents and increase the proportion of relevant documents at top ranks. Initial experiments using manually formulated filters yielded some improvement, but did not outperform the automatic QC algorithm described above. Further, the automatic generation of appropriate Boolean filters given a natural language query is a difficult task, and automatic filters are expected to do worse than the manual filters used in our experiments. Thus, this approach was also not pursued for our final run.

Improved expansion. We tried using document clustering as well as term clustering to improve the expansion step. These approaches were also motivated by query coverage — we would like to ensure that multiple key concepts in the user query are nicely represented in a balanced way in the expanded query.

When the top-ranked documents for a query are clustered, the different clusters often correspond to different key concepts. For example, for TREC query 203 (*What is the economic impact of recycling tires?*), the top documents could form clusters corresponding to *recycling* (discussing garbage, recycling of plastic, tin cans, etc.), *tires* (dealing with rubber, automobile tires, the Goodyear company, etc.), and *economy* (dealing with financial matters). The expanded query for such a topic should include terms from each of these different classes. Otherwise, the expanded query could be dominated by one concept, e.g. recycling, and would then retrieve many non-relevant documents, dealing for example, with the recycling of tin cans.

Even when many of the top-ranked documents are relevant and cover most of the concepts in the query, clustering the top documents and selecting terms from each cluster should be useful, since it would ensure that the expanded query addresses different kinds of relevant documents rather than becoming a one-sided representation of the search topic that would find relevant documents of a particular kind only.

There are some queries, however, for which particular types of non-relevant documents also cluster. For such queries, selecting terms from all clusters would introduce a bias in the query in favor of these non-relevant documents, and performance would deteriorate. Ideally, we would like to be able to select terms from only those clusters that contain useful terms (usually clusters that contain a large proportion of relevant documents). Since we do not know this information *a priori*, we use the following heuristic to select clusters for expansion. Cluster vectors are compared to the original query, and ranked in order of similarity. Terms from the best two clusters are used for query expansion. Hopefully, this method would retain most of the benefits of clustering for queries where it helps, while minimizing the damage for the kind of queries mentioned above where clustering hurts performance.

The actual clustering algorithm used is straightforward. Clusters are initialized with one document

per cluster. A *cluster vector* is associated with each cluster. Initially, this vector is simply the vector corresponding to the single document contained in that cluster. The similarities between pairs of clusters are computed using the inner product similarity between the corresponding vectors. The most similar pair of clusters is merged into a single cluster by combining all the documents in the two clusters into one large document, and similarities between clusters are recomputed. The merging process continues until the similarity between the most similar pair of clusters falls below a threshold.

Combining these techniques, our final strategy for the first official run (Cor6A1cls) is the following:

1. Retrieve 1000 documents using the initial query (using *Lnu.ltu* weights).
2. Generate cooccurrence information about the query terms from the top 1000 documents.
3. Rerank the top 50 documents as in TREC 5 (using correlation and proximity information).
4. Assume the top 20 documents relevant, documents ranked 501–1000 non-relevant.
5. Generate clusters for the top 30 documents and save the best (most heavily weighted) terms from each cluster vector.
6. Rank the cluster vectors according to their similarity to the original query (using *bnn* weights for the clusters) and select the best 2 clusters.
7. Expand the query by 100 words and 20 phrases using $\alpha = 8$, $\beta = 8$, and $\gamma = 8$. The expansion terms are selected from among the saved terms for both clusters and the actual number of terms selected from a cluster is proportional to its similarity to the original query.
8. Retrieve the final set of 1000 documents using the expanded query.

SuperConcepts.

It was clear from tuning runs that the cluster approach above was having some effect, but not as much as we hoped query balancing would get. So we developed a last-minute experimental approach, SuperConcepts, that directly attacks the problem of query balance while expanding a query. The first working implementation was run the day before TREC results were due, so not much tuning was done, or has been done since. Hopefully we will have fuller results to report by the time of the workshop itself.

The goal of SuperConcepts is to balance the aspects of an original query given any reasonable expansion of that query. In the sample query from above (*What is the economic impact of recycling tires?*), an expansion of it may add 20 terms dealing with *economic* but only 5 terms related to *recycling tires*. Unless we balance the expanded query, we may retrieve only financial documents.

The basic approach is to create SuperConcepts of related terms.

1. Assign each original query term to be the seed of a Superconcept.
2. Assign every expansion term to every correlated SuperConcept seed, dividing its feedback weight proportionally to the correlation.
3. Finally, apportion part of each SuperConcept to other more highly weighted correlated SuperConcepts.

For example, suppose we have an expanded query of (⟨ economic,.6 ⟩ ⟨ recycling,.8 ⟩ ⟨ tires,.9 ⟩ ⟨ financial,.5 ⟩ ⟨ profit,.1 ⟩ ⟨ loss,.1 ⟩ ⟨ Goodyear,.3 ⟩) where the first three terms were the original query terms. Then we might end up with SuperConcepts of

- ⟨ economic,.6 ⟩ ⟨ financial,.5 ⟩ ⟨ loss,.1 ⟩ ⟨ profit,.08 ⟩ ⟨ Goodyear,.05 ⟩

- ⟨ recycling,.8 ⟩ ⟨ profit,.02 ⟩
- ⟨ tires,.9 ⟩ ⟨ Goodyear,.25 ⟩

where *Goodyear* has been allocated mostly to the *tires* SuperConcept, but a bit to the *economic* SuperConcept.

A SuperConcept is matched against a document by matching against the included terms. But rather than including each match as an independent inner product match, we deprecate the importance according to how many other terms of this SuperConcept have matched. More precisely, we sort the SuperConcept terms by decreasing weight, and then match each term in turn. The contribution of a single term match is defined to be

$$1/(1 + cn) * qwt * dwt \tag{2}$$

where c is a constant, n is the number of terms that have previously matched this SuperConcept, and dwt and qwt are the document and query weights of this term.

In the SuperConcept tire example above, if c is 1, then a document matching

- *economic* will have an effective query weight of .6
- *financial* will have an effective query weight of .5
- both *economic* and *financial* will have an effective query weight of .85 (.6 + 1/2 .5).

Thus each additional match of a term related to *economic* will count a bit less, just as multiple matches of the same term normally count as the *log* of the number of matches. Note that $\int 1/(1+n) = \log(n)$.

The final SuperConcept approach is

1. Use a base approach to determine an expanded query
2. Form SuperConcepts from both the original query and the expanded query.
3. Match SuperConcepts against documents, which deprecates multiple terms matching within the same SuperConcept.

Ad-Hoc experiments and analysis

We submitted three runs in the ad-hoc category: Cor6A1cls starts with the description-only queries and uses document clustering during expansion; Cor6A2qtcs clusters the terms in an initial TREC 5 style expanded query into SuperConcepts and uses the SuperConcepts for the final retrieval; Cor6A3cll is identical to Cor6A1cls except that it starts with the full queries instead of the description field only.

Table 1 shows the results for the various runs across 50 queries. The performance level for the short queries is fairly poor in absolute terms. Using full queries improves performance by about 20%. Due to an indexing error, the queries used in our official Cor6A3cll run ontained the description and narrative fields only, but not the title. (NIST inadvertently changed the format of the title field of the topic. Since all of our processing is automatic, we didn't notice.) We reran this run after fixing the problem and obtained a 13% increase over our official average precision figure. This is somewhat surprising in view of the brevity of the title (which usually consists of only 2–3 words), but given that this field contains the essence of the query (in contrast to the description and narrative sections which often contain extraneous terms) this improvement is reasonable.

Table 2 shows that our runs compare reasonably with other runs. For several queries, however, our short-query runs do not work well — the performance of these runs falls below the median on 13 queries. The results for the long queries are somewhat better. The incorrect run is above median on 40 queries and is

Run	Average precision	Total rel retrieved	R precision	Precision @100 docs
Cor6A1cls	.1799	2391	.2155	.1794
Cor6A2qtcs	.1809	2332	.2076	.1714
Cor6A3cll	.2139	2590	.2415	.2010
Cor6A3cll (fixed)	.2413	2852	.2650	.2242

Table 1: Ad-Hoc results (50 queries)

Run	Task pool	Best	$\geq$ median
Cor6A1cls	Short automatic	0	37
Cor6A2qtcs	Short automatic	0	37
Cor6A3cll	Long automatic	2	40
Cor6A3cll (fixed)	Long automatic	2	44

Table 2: Comparative automatic ad-hoc results (50 queries)

best on 2. For the fixed run, these numbers are 44 and 2 respectively. (Note that these are only approximate numbers for the fixed run, since the statistics computed from the pool of runs change if the incorrect run is replaced by the correct one.)

We analyze our first run in greater detail in Table 3. The base vector run using single terms only yields a low average precision of 0.1479. Using phrases in the initial retrieval improves performance by almost 8%. Reranking the top 50 documents using cooccurrence and proximity information improves results by another 4%[1]. When the terms in the original vector are reweighted using the assumed relevance information, we get a further improvement of 5%.

Run	Avg. P
1. Vector (*Lnu.ltu*), single terms only	1479
2. above + phrases	1593 (+7.7%)
3. After reranking top 50	1648 (+11.4%)
4. Reweighted vector (no expansion)	1728 (+16.8%)
5. Expansion (no reranking, no clusters)	1831 (+23.8%)
6. Exp (reranking, no clusters)	1804 (+22.0%)
7. Exp (no reranking, clusters)	1825 (+23.4%)
8. Cor6A1cls (reranking, clusters)	1799 (+21.6%)

Table 3: Ad-Hoc component results (50 queries)

The last four rows in Table 3 attempt to analyze the relative importance of the two principal ingredients of our approach — *reranking* to improve the set of 20 documents that are assumed relevant, and *clustering* to ensure that the final query is a well-balanced one. Unfortunately, the numbers are fairly close to each other and it is difficult to draw reliable conclusions from them. The simplest approach that uses neither reranking nor clustering seems to work best for the TREC 6 task, unlike earlier tasks.

A look at the results for individual queries shows that each technique helps and hurts performance on approximately the same number of queries. For example, comparing the runs corresponding to lines 5 and 6, we find that reranking significantly improves performance (by at least 8%) on 17 queries, but hurts performance on 19. Similarly, comparing the runs corresponding to lines 5 and 7, we find that clustering improves the results for 10 queries but hurts performance for 7. We need to analyze the results in greater detail to come to a conclusion about the reliability and stability of our methods.

We also study the effect of query length on retrieval effectiveness for the TREC 6 task. Table 4 shows the average precision and total number of relevant documents retrieved when various sections of the query are indexed. The base run retrieves 1000 documents per query using just the vector match. The final run starts with the appropriate set of indexed queries but is otherwise identical to Cor6A1cls.

[1]Note that, for consistency, we evaluate this run by computing the average precision at 1000 documents. The benefit of reranking the top 50 documents is therefore partly concealed by the unchanged ranks of the remaining 950 documents and is more significant than suggested by the 4% improvement.

Indexed sections	Title only	Desc only	Title+Desc	Full
Base run	0.1959	0.1593	0.2040	0.2169
	2200	2031	2369	2522
Final run	0.2169	0.1799	0.2306	0.2413
	2483	2391	2680	2852

Table 4: Ad-Hoc results for various query lengths (50 queries)

Traditionally, longer queries have yielded better results. The performance trends shown in Table 4 generally agree with this observation, with one significant exception. The description-only queries do rather badly compared to the extremely short (often single-word) title-only queries. This can be explained as follows. The title section contains the word(s) most crucial to the query, whereas the description is often more verbose and sometimes states minor constraints that have to be satisfied by relevant documents. This introduces extraneous terms which dilute the importance of the core concepts and lead the retrieval process astray.

It also appears that some queries may have been designed by the assessors with the idea that both the title and the description would be used. The description field very often uses an alternative vocabulary to describe the concept from the title. This makes sense for a user to do if the two fields are to be used together, but hurts if the fields are to be considered separate queries.

Routing

Routing Methodology

Continuing the Cornell tradition, we submitted one "conservative" run based on previously developed and well-tested techniques (Cor6R1cc), and one experimental run (Cor6R2qtc) based on recent work.

The Cor6R1cc run uses the same techniques as our TREC 5 routing submission (Cor5R1cc). The development of this approach is presented in detail in [5]. Below, we briefly recapitulate the steps involved in generating the routing queries:

1. Create the initial query vector with *ltu* weights. Inverse document frequency information is obtained from the training collection.
2. For each query, retrieve the top 5000 documents from the training set.
3. Expand the query by adding single terms and phrases that occur in more than 5% and 10% resp. of the relevant documents.
4. Weight the terms in the expanded query using the Rocchio feedback formula. Only those non-relevant documents that are within the query zone are used in this step.
5. Add pairs of cooccurring terms that occur in more than 7% of the relevant documents.
6. Compute weights for the added pairs using Rocchio's formula. Only the top-ranked $2R$ non-relevant documents (where R is the number of relevant document for the query) are used in this step.
7. Retain the most highly weighted 100 single terms, 10 phrases and 50 cooccurrence pairs in the final query.
8. Run the expanded query through a 3-pass Dynamic Feedback Optimization (DFO) step to fine-tune the weights.

The experimental SuperConcept run, Cor6R2qtc, directly takes the expanded query used in Cor6R1cc, and assigns the expansion terms to SuperConcepts seeded by the original query. This is the first, and so far only, routing run ever done with SuperConcepts; we had no time to tune it.

Routing automatic results and analysis

The performance figures for the official Cornell submissions are shown in Table 5. Both the runs are in the automatic category. We recently discovered an error in our indexing script for the test database. The results obtained when this error is corrected (and a few other minor changes made — certain document sections that were being previously omitted are indexed) are shown in the last row. The absolute figures are good, though it is perhaps somewhat surprising that we are unable to do better in spite of the enormous quantity of training data.

Run	Average precision	Total rel retrieved	R precision	Precision @100 docs
Cor6R1cc	.3983	5429	.4198	.3930
Cor6R2qtc	.3766	5233	.3999	.3802
Cor6R1cc (fixed)	.4028	5483	.4174	.3949

Table 5: Automatic routing results (47 queries)

Run	Best	$\geq$ median
Cor6R1cc	3	47
Cor6R2qtc	1	41
Cor6R1cc (fixed)	1	45

Table 6: Comparative automatic routing results (47 queries)

Table 6 compares our submissions to other submissions in this category. The performance of our methods is impressive — Cor6R1cc performs at or above the median on all queries and is best on 3. Cor6R2qtc is only somewhat less consistent with a performance below the median on 6 queries.

The Cor6R2qtc result is disappointing but not too surprising. The DFO optimizes the weights assuming an inner product similarity function. Those weights will not be optimal when distributed across SuperConcepts. We have an alternative implementation of DFO that is valid for SuperConcept matching almost done; we hope to have figures for that run by the workshop.

We show the step-by-step analysis of Cor6R1cc in Table 7. The basic vector run starts at an average precision of 0.2640. Simply reweighting the original query terms using Rocchio's formula yields a 14% improvement. Expanding by 100 terms and 10 phrases gives us a considerable improvement of 18%. Adding pairs of cooccurring words improves results marginally, and the final significant improvement in performance comes from optimizing the query term weights. The contribution of each step to the final performance on this task is very similar to that for the TREC 5 task [5]. This is reassuring in terms of the stability of our techniques.

Run	Avg. P
1. Vector (*Lnu.ltu*), incl. phrases	2640
2. Reweighted vector (no expansion)	3010 (+14.0%)
3. Expansion by 100 terms, 10 phrases	3485 (+32.0%)
4. Exp by 100 terms, 10 phrases, 50 pairs	3646 (+38.1%)
7. Above + DFO (Cor6R1cc)	3983 (+50.9%)

Table 7: Routing component results (47 queries)

High-Precision

The TREC 6 High-Precision track is a new track this year. It is an attempt to perform a task that is much more closely related to real-world user interactions than the ad-hoc or routing task. The goal is simple: the user is asked to find 10 relevant documents in 5 minutes. No other restrictions are put on the user (other than no prior knowledge of the query, and no asking other users for help). Evaluation is simply how many actual relevant documents were found among the 10 documents supplied by the user (Precision at 10 documents).

High-Precision Methodology

Our methodology for the TREC 6 high-precision task is very similar to the one we adopted for the TREC 4 Interactive and TREC 5 Manual ad-hoc runs [4, 5]. The user's main task is to provide relevance judgements to be fed to our standard Rocchio relevance feedback algorithm. Direct modification of the query (adding/deleting terms to/from the query) was also occasionally (rarely) used by the searchers. The other principal component of our technique, is the use of pipelining or "parallel" processing so that expensive retrieval techniques can be executing while the user continues to make judgements. The details of the method are given below:

1. The current time is noted. The user views the query supplied by NIST and enters it, either as-is or suitably modified, into the system.

2. The query entered by the user is indexed and a set of documents is retrieved using a simple vector match.

3. The top-ranked documents are presented to the user.

4. The user starts viewing the documents and judging them 'relevant', 'non-relevant' or 'possibly relevant'.

   In parallel, a child process is forked to retrieve additional documents using a more sophisticated retrieval algorithm: the initial query is used to retrieve 1000 documents, the top 20 are assumed to be relevant, documents ranked 501-1000 are assumed to be non-relevant, and automatic feedback is used to expand the query by 25 single terms and 5 phrases, using $\alpha = 8, \beta = 8$ and $\gamma = 8$. The expanded query terms are then grouped into superconcepts and this query is run to retrieve the final set of documents. This run corresponds to our official Cor6A2qtcs run.

5. After every judgment, the current time is noted. All documents retrieved so far are sorted such that the documents judged relevant come first, followed by all documents judged possibly relevant, followed by all unjudged documents, and the top 10 documents in this ranking are saved in a file and time-stamped.

6. After every 5 categorical judgments (i.e. 'relevant' or 'non-relevant'), a relevance feedback process is started in parallel if the child process is idle. For this process, documents marked relevant by the searcher are assumed to be relevant, and documents marked non-relevant as well as those retrieved at ranks 501-1000 by the initial user query are assumed to be non-relevant. Documents marked "possibly relevant" are not used in the feedback process. The query is expanded by 25 words and 5 phrases. $\alpha = 8, \beta = 8$ and $\gamma = 8$ are used. While this feedback process is running in the background, the user continues to judge more documents.

7. When the child process is done (i.e. retrieval or feedback completes), and the new retrieval results are available, these results are merged into the current list of top-ranked documents being shown to the user.

8. The final top 10 documents for the query will be the last set of 10 documents saved with a timestamp under the 5-minute limit

User Interface. The user interface for the TREC 6 high-precision runs is a simple textual interface that does not use any windowing[2]. The UI is used to view documents and mark documents 'relevant', 'non-relevant', or 'possibly relevant'. Query term occurrences in document texts can be optionally highlighted. The interface also displays the time elapsed since the beginning of the search. The interface may also be used to modify the query as follows: the text of the current query is shown to the user who makes appropriate changes and submits the modified query, which is then used in all subsequent processing.

Users. Three runs are presented; each the result of one user running all 50 queries. The user and some environmental characteristics are:

[2] It is very similar to the interface used in the TREC 4 interactive task. This interface is described in detail in [4]

1. User 1 - Run HP1
    - Experience: System designer (HP interface designer)
    - Machine: Sparc 20/512 (old low end machine)
    - UI Settings: Highlighting of terms on
2. User 2 - Run HP2
    - Experience: SMART System implementer
    - Machine: UltraSparc 140 (new low end machine)
    - UI Settings: Highlighting of terms off
3. User 3 - Run HP3
    - Experience: System designer
    - Machine: UltraSparc 140 (new low end machine)
    - UI Settings: Highlighting of terms off

All users should be considered experts although User 2 had much less SMART and TREC experience. User 1 was running on a slower machine (by a factor of 2), which undoubtedly had an impact on how many of the more expensive runs finished. Users 2 and 3 decided to turn highlighting of document terms that occurred in the query off. On longer documents, highlighting was expensive (2–3 seconds) and it was not felt it was worth it.

The evaluation results are presented in Table 8. The base case is the official run Cor6A3cll which gives the precision at 10 documents of that automatic run. The Cor6HP3 run got close to 2/3 of the optimum performance. It also was greater than or equal to the median on 84% of the queries, being the best (or tied for best) for 36% of the queries.

Run	Precision	Relative Precision	Num queries Best	Num queries $\geq$ Median
Base	.4260	-	-	-
Cor6HP1	.5540	.5564	10	38
Cor6HP2	.5660	.5820	12	39
Cor6HP3	.6020	.6298	18	42

Table 8: High-Precision comparison (50 queries)

One important question is how the users agreed with the official TREC relevance judgements. If the HP track is to have meaning, the disagreement between user interpretation of relevance to a query, and the official assessor interpretation can't dominate the results. Table 9 gives the total number judged relevant, possibly relevant, and non-relevant for each user, along with the corresponding number officially judged relevant. For example, of 690 documents judged non-relevant by User 3, 58 (8%) had been judged relevant by the assessors. The final column shows the number of documents judged non-relevant or possibly relevant for which our trace did not record the docid. For these documents, we cannot tell whether they were officially relevant or not.

In general, from 68% to 77% of the documents judged relevant by the users were judged relevant by the assessors. This agrees with the previous TREC consistency studies done by NIST. The disagreements tended to concentrate on only a few queries. 3/4 of the queries had 0 or 1 disagreements on the user judged relevant documents. But, for example, on query 305 the three users found a total of 24 documents they thought were relevant, with 6 more possibly relevant. Of those 30 documents, none(!) were judged relevant by the assessor. Query 345 had 19 disagreements, and query 309 had 15 disagreements.

Examining the figures in Table 8 and Table 9 more closely, there is an apparent correlation between overall effectiveness and number of documents judged. User 1 (on the slow hardware with highlighting) only judged 1029 documents, while User 2 judged 1040 and User 3 judged 1360 documents.

Run	User judged Relevant	Official Relevant	User judged Possibly Rel	Official Relevant	User judged NonRel	Official Relevant	Non-relevant Unknown docs
Cor6HP1	300	225	111	40	430	33	188
Cor6HP2	398	270	15	3	527	48	140
Cor6HP3	345	265	103	32	690	58	162

Table 9: High-Precision User-assessor consistency (50 queries)

Being on a slower machine caused User 1 problems, in that the more expensive (and hopefully better) iterations did not finish as quickly as for the other users. Even on the faster machine, speed was somewhat a problem. User 3 took that into consideration and used shorter initial queries, averaging 22 words/phrases per query instead of 30 for User 1 and 32 for User 2. The shorter queries shortened the time for the more expensive iterations and thus allowed them to be used for more judgements. Thus User 3 was able to judge 358 documents that were retrieved as the result of feedback (iterations 2-7), while User 1 only examined 24. Table 10 gives the number of retrieved documents per iteration, along with the number judged relevant.

Run	Rel/Ret Iter 0	Rel/Ret Iter 1	Rel/Ret Iter 2	Rel/Ret Iter 3	Rel/Ret Iter 4	Rel/Ret Iter 5-7
Cor6HP1	236/903	59/102	0/12	5/12	-	-
Cor6HP2	235/719	97/178	52/99	10/33	2/17	2/3
Cor6HP3	165/703	75/299	72/187	25/102	7/48	1/21

Table 10: High-Precision Relevant/Retrieved for each Iteration

Overall, the high-precision track itself seemed to work. Consistency of judgements did not seem to be much of a problem. The range of evaluation results appears to be good. If the runs were much better (in the 75% precision range, instead of 60%), then the consistency issues may start dominating the comparisons between runs. At least for our runs, the results are very understandable: if you look at more good documents, you get better results!

Cross-lingual

Once again, our emphasis in our multi-lingual runs is to see how effective retrieval can be with a minimal amount of linguistic information. Good linguistic information should someday be able to improve a good non-linguistic (statistical) search. However, prematurely concentrating on the linguistic aspects of operating on different languages may not only yield sub-par retrieval, but indeed may interfere with evaluation of the linguistic approach. A useful linguistic technique tied to a sub-optimal statistical base retrieval may be unfairly judged as not important.

As we did in TREC 3 with Spanish, we spent a small amount of time determining simple stemming rules for French and coming up with a French stopword dictionary. We use these to perform a mono-lingual French-French run, using almost exactly the same technique as for our English SuperConcept run, Cor6A2qtc. (These cross-lingual runs are described in more detail in later sections.)

We use the same document collection for an English-French cross-lingual run. No dictionaries are used; instead the English query words are treated as potentially mis-spelled French words. The English query is expanded by adding French words from the collection that are lexicographically nearby. This query is then run as a normal mono-lingual run (including more automatic expansion using an initial retrieved set).

Another "cross-lingual" run is English queries against machine translated German documents. Exactly the same techniques as a pure English mono-lingual run are applied.

Finally, we also did a base case English-English run.

French run preparation

The French collection was indexed using a combined French/English stopword list and a French stemmer. The English stopwords were the SMART default list of 500-some; about 300 entries were added for French,

starting by going over the 1000 most frequent words from one year of the French collection and adding any pronouns, articles, or common tenses of être and avoir that hadn't already shown up, along with a few translations of English stopwords such as numbers. (An exception was été, which was not put in as a stopword, although être uses probably overwhelm those for summer.)

While examining the most frequent words from the French collection, it became apparent that a large minority of words that should have had accent marks did not. We decided to drop all accent marks that did exist during indexing to get these different representations of the same word to match each other.

The French stemmer removed *l'*, *d'*, *s'*, *n'*, *qu'*, *j'*, and *m'* from the beginning of words. Final "ment" was removed in hopes of matching adverbs and adjectives. Many verb forms were returned to the infinitive by trimming final "erai", "erons", "erez", "eront", "erais", "erait", "erions", "eriez", and "eraient" to "er", and similarly for the "ir" forms. (Final "era" and "eras" were left alone as occurring too often outside verb forms and not that often as verbs.) A few very rough attempts were made to deal with masculine/feminine and singular/plural, with final "elle", "elles" being trimmed to "el"; "enne", "ennes" to "en"; "if", "ifs" to "iv" (matching trimmed "ive" and more English words); and a fallback removal of final "s", "e", "es", and "'s" (the last due to apparent English contamination in the collection).

Mono-lingual runs

For the French-French mono-lingual run, the French queries are run through the above process before being matched against the French documents. Then the techniques used in our English SuperConcept ad-hoc run, Cor6A2qtc, are applied to the French collection. The only difference is that no phrase indexing is done. The run does very well when evaluated on the 13 queries for which relevance judgements are available (Table 11). It is greater than or equal to the median run on all 13 queries, and best on one query.

Run	Average precision	Total rel retrieved	Num queries Best	Num queries $\geq$ Median
Cor6FFsc	.3994	585	1	13

Table 11: French-French monolingual run (13 queries)

Our English-English mono-lingual run is a base case for other systems to compare against. This is exactly the Cor6A2qtc ad-hoc run except with the cross-lingual English queries and documents. One thing to note about the results in Table 12 is the absolute level of performance. The average precision is more than twice as high as the ad-hoc run. This should help allay some fears about the level of ad-hoc performance; when put in a comparatively easy environment (easy queries and smaller collection), the absolute level of performance is good.

Run	Average precision	Total rel retrieved	Num queries Best	Num queries $\geq$ Median
Cor6EEsc	.4568	791	5	12

Table 12: English-English monolingual run (13 queries)

English-French runs

The French/English cross-language retrieval was based on the idea that the languages share many, many cognates. English swallowed much of French vocabulary (although not grammar) quite recently as language developments go. Lists of the most frequent non-stopword terms from English and French news collections suggested that 10-15% of terms would, modulo stemming, match exactly across languages, while another 25-30% were close enough that there was a reasonable hope that they could be matched automatically.

A trie was built of all indexed terms from the French collection occurring at least five times, in the hopes that we could massage English cognates to match these French terms. Terms extracted from the English queries, using the standard English stemmer, were run against the trie, and French terms found were added to the query. Besides the general procedures of adding or deleting one letter, two equivalence

classes of letters were defined after studying the frequent English and French terms. One equivalence class was vowels, such that an English term would pick up French terms with the same consonant pattern but different vowel sequences. The other was k sounds, where any combination of c-k-qu could substitute for any other. Naturally, some unwanted terms were added at this stage, but so were a large number of cognates.

The next step to improve retrieval was to automatically expand the query by adding terms occurring in the top 20 ranked documents (just as we do in pure English retrieval). We assume that correct cognates are sufficient to ensure the top documents are in the general subject area of the query. The expansion terms from the top documents should give us related pure French non-cognates. The new terms were weighted according to our normal Rocchio formulation, and the new query was then run against the collection again to give us our final results. The SuperConcept approach used in the mono-lingual run was not used here, though in future runs it could be.

In one of the runs, a small thesaurus of about 300 words (half time and geographic references, half common or important words that were not cognate between the languages) was used to automatically add additional French terms to the query. This thesaurus was prepared from general knowledge before the queries were known. Using the thesaurus increased recall in several cases, but had no particular effect on precision.

This very simple, non-linguistic approach did very well! As Table 13 shows, the non-thesaurus run was above the median for 10 out of the 13 queries, with a quite respectable average precision, about 60% of the very good mono-lingual run.

Run	Average precision	Total rel retrieved	Num queries Best	Num queries $\geq$ Median
Cor6EFent	.2408	407	2	10
Cor6EFexp	.2435	421	1	11

Table 13: English-French cross-lingual run (13 queries)

Query-by-query results comparing the mono-lingual run against the cross-lingual run were about as would be expected from the discussion above. For queries #1 (Waldheim/Nazi/crime) and #14 (international terrorism(e)), the terms came out the same in both languages. For query #6, the terms were similar in both languages, but the English query happened to use "air" while the French query used "atmosphère", causing the English query to do better because more documents talked about air pollution than atmospheric pollution. For queries #5 (medicine vs. médecine) and #10 (solar vs. solair(e)), vowel substitution worked nicely. For query #9, main terms like "deforestation" matched, but others like "flood"/"inondation" did not, depressing results from the French query. For query #19, "wine" and "vin" did not match, although that would be within the abilities of a more robust cognate-finder. For queries #17 (potato vs. pomme de terre) and #24 (teddy bear vs. ours en peluche), there was no hope for an automatic match without a major dictionary, although "teddy bear" did occur twice in the French documents.

In this initial investigation, we used a human to come up with the rules for what extra types of "misspellings" should be considered (i.e., vowel substitution and c-k-qu equivalence). However, there is no reason why these rules cannot be learned automatically for any pair of related languages. Just trying all transformations of words in one language, and seeing which transformations often end up with a word in the other language should work. This can be explored in the future.

The approach used here is only useful for related languages. We need to discover how related the languages need to be for reasonable performance. There are large numbers of former colonies in Africa, the Carribean, and elsewhere, whose everyday language has drifted from that of the former colonial power. This result suggests that we do not need to consider those as separate languages, with distinct linguistic support needed for retrieval.

English-TranslatedGerman runs

Our English-TranslatedGerman run is just a quick run to see what happens if we treat it entirely as a mono-lingual English run. We did no analysis of any factors that might make the translated environment different from normal English, and just ran our standard English ad-hoc SuperConcept run. Table 14 shows that we

are above the median for 11 out of the 13 queries, though the absolute level of performance does not seem very exciting.

Run	Average precision	Total rel retrieved	Num queries Best	Num queries $\geq$ Median
Cor6ETGsc	.1858	393	1	11

Table 14: English-TranslatedGerman cross-lingual run (13 queries)

Chinese

Last year we participated in the first Chinese track and had one of the very top results. It was a very low effort track for us; nobody from our group understood any Chinese at all and we had no training data, so all we could do was run our basic approach treating single Chinese characters as words, and pairs of Chinese characters as phrases.

This year we still have no one who understands any Chinese, but we have last year's work as training data, so the effort involved has been a bit greater.

Unfortunately, most of our simple attempts to improve last year's statistical results have very little improvement. The final official submissions are based on this year's English ad-hoc run using SuperConcepts, Cor6A2qtc. Official Chinese run Cor6CH1sc follows exactly the same procedure as the English run, except it was decided to treat the two-character phrases as being the base concepts of the SuperConcepts instead of the single terms as in the English run.

The second official run, Cor6CH2ns, is exactly the same as Cor6CH1sc, except no expansion single characters were added. Instead of adding 15 single terms and 15 phrases to the original query from the top 20 initially retrieved documents, only 25 phrases were added.

Table 15 shows the results. There is disappointingly little difference between the two runs. Both runs did very well, with the Cor6CH2ns being greater than or equal to the median on 18 out of the 26 queries.

The absolute performance level achieved by everybody is extremely impressive. The median performance level (averaging the median average precision for all queries) of .535 is by far the highest level of performance of any TREC track in history. However, it remains unclear why this level of performance is being achieved. Is it a property of the Chinese language, perhaps due to less ambiguity of important terms? Or is it a property of these particular Chinese queries, prepared by assessors who know the vocabulary used in the target document set very well?

Run	Average precision	Total rel retrieved	Num queries Best	Num queries $\geq$ Median
Cor6CH1sc	.5547	2765	0	16
Cor6CH2ns	.5552	2763	0	18

Table 15: Chinese automatic ad-hoc (26 queries)

Comparison with past TREC's

We will present our standard comparisons with previous TREC's at the workshop; it is unfortunately not completed yet.

Conclusion

The Cornell SMART Project is again a very active participant in this year's TREC program. With the exception of the high-precision relevance feedback runs, everything we have presented here is completely automatic and uses no outside knowledge base (other than a small list of stopwords to ignore while indexing). Manual aids to the user can be built on top of this system to provide even greater effectiveness.

We use two approaches, clustering and SuperConcepts, to try to preserve query balance while expanding a query for the ad-hoc task. Both have been mildly successful on other TREC collections (3% – 7% improvement) but help only marginally, if at all, on the main TREC ad-hoc task (Description field only queries). One explanation for this could be that the Description field is not intended to be a stand-alone query, but intended to be used in conjunction with the title field. (The very short title field perform significantly better by itself than the longer description field.)

For routing, we unsuccessfully tried to use SuperConcepts to balance the query. It is obvious more work needs to be done there; because of time pressure, no tuning or trials were done before the official run. We also submitted an official run using our TREC 5 routing approach; that run is still one of the best runs, doing better than the median for all queries.

Our entries in the new High-Precision track are based almost entirely on relevance feedback. Our users read documents and judge them, rather than attempting to manually reformulate a query, or use query visualization techniques to focus in on relevant documents. Results are much better than the median.

Little new work was done for the Chinese track. Results are better than the median for 2/3 of the queries; a nice result considering nobody in our group understands a word of Chinese.

Our French-French mono-lingual run is very successful, above the median on all queries. The only language related work done for this run was the construction of simple stemming rules and a simple stopword list.

Our English-French cross-lingual is surprisingly successful. We use almost no linguistic information, and just treat the English query terms as mis-spelled French words that need to be corrected. This suggests that retrieval across related languages does not need a lot of apparatus to be effective.

References

[1] Chris Buckley, James Allan, and Gerard Salton. Automatic routing and ad-hoc retrieval using SMART : TREC 2. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 45–56. NIST Special Publication 500-215, March 1994.

[2] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[3] Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using SMART : TREC 3. In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*. NIST Special Publication 500-225, 1995.

[4] Chris Buckley, Amit Singhal, and Mandar Mitra. New retrieval approaches using SMART : TREC 4. In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*. NIST Special Publication 500-236, 1996.

[5] Chris Buckley, Amit Singhal, and Mandar Mitra. Using query zoning and correlation within SMART : TREC 5. In D. K. Harman, editor, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-???, 1997.

[6] Mandar Mitra, Chris Buckley, Amit Singhal, and Claire Cardie. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO '97*, 1997.

Appendix A - Cross-lingual Questionnaire

1. OVERALL APPROACH:

1.1 What basic approach do you take to cross-language retrieval?
[ ] Query Translation
[ ] Document Translation
[X] Other, ''English is merely mis-spelled French''

1.2 Were manual translations of the original NIST topics used as a starting point for any of your cross-language runs?
[X] No
[ ] Yes, _____

1.3 Were the automatically translated (Logos MT) documents used for any of your cross-language runs?
[ ] No
[X] Yes, Did an English against translated German run

1.4 Were the automatically translated (Logos MT) topics used for any of your cross-language runs?
[X] No
[ ] Yes, _____

3. USE OF MANUALLY GENERATED DATA RESOURCES:

3.1 What kind of manually generated data resources were used?
[ ] Dictionaries
[ ] Thesauri
[ ] Part-of-speech Lists
[X] Other: we developed very simple French stemmer, stopword list

3.2 Were they generated with information retrieval in mind or were they taken from related fields?
[X] Information Retrieval
[ ] Machine Translation
[ ] Linguistic Research
[ ] General Purpose Dictionaries
[ ] Other, _____

3.3 Were they specifically tuned for the data being searched (ie. with special terminology) or general-purpose?
[ ] Tuned for data; Please specify _____
[X] General purpose

3.4 What amount of work was involved in adapting them for use in your information retrieval system.
[ ] None

[X] Developed from scratch in about 6 hours.

3.5 Size
[X] 879 stopword entries
[ ] 0.005 MBytes

3.6 Availability? - Please also provide sources/references!
[ ] Commercial
[ ] Proprietary
[X] Free
[ ] Other, _____

4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES:

4.1 Form of the automatically constructed data resources?
[ ] Lexicon
[ ] Thesaurus
[ ] Similarity matrix
[ ] Other
[X] None

4.2 What sort of training data was used to construct them?
[ ] Same data as used for searches, _____
[ ] Similar data as used for searches, _____
[ ] Other data, _____

4.3 Size
[ ] _____ entries
[ ] _____ MBytes

4.4 Was there any manual clean-up involved in the construction process?
[ ] Yes, _____
[ ] No

4.5 Rough resource estimates for building the data resources (ie. an indicator of the computational complexity of the process).
[ ] _____ hours
[ ] _____ MBytes of memory used
[ ] _____ temporary disk space

5. GENERAL

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?
[ ] Very dependent, _____
[ ] Somewhat dependent, _____
[ ] Easily replacable, _____
[ ] Don't know
[X] No significant data resources used.

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?
[ ] Yes, a lot, _____
[ ] Yes, somewhat, _____
[X] No, not significantly, _____
[ ] Don't know

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?
[ ] Yes a lot, _____
[ ] Yes, somewhat, _____
[X] No, not significantly, _____
[ ] Don't know

5.4 Are similar resources available for other languages than those used?
[X] Yes, Approach only valid for languages with some similarity.
[ ] No

Okapi at TREC–6
Automatic ad hoc, VLC, routing, filtering and QSDR

S. Walker* S.E. Robertson* M. Boughanem* G.J.F. Jones† K. Sparck Jones‡

Advisers: E. Michael Keen (University of Wales, Aberystwyth), Karen Sparck Jones (Cambridge University), Peter Willett (University of Sheffield)

Note on notation

In the tables **P<n>** means precision at cutoff <n> documents and **RPrec** means precision after R documents have been retrieved, where R is the number of known relevant documents. *TSV* means Term Selection Value (see Section 3.1).

1 Introduction

Automatic ad hoc

Many experiments were concerned with "blind" expansion (i.e. expansion using pseudo-relevant and -nonrelevant documents). A very large number of runs were done on TREC–3, 4 and 5 data to investigate the effect of varying the Okapi BM25 parameters on precision at low recall. Methods of selecting and ranking topic terms and expansion terms were investigated. In particular, introducing a "non-relevance" component into the expansion term selection function appears to give a small benefit. This work produced good results on TREC-5 data.

Three runs were submitted: long, description, and title only. There was a mistake in the long run. With this corrected, all three runs were among the best, on most statistics.

VLC track

We were interested to find out whether the Okapi BSS (Basic Search System) could handle more than 20 gigabytes of text and 8 million documents without major modification. There was no problem with data structures, but one or two system parameters had to be altered. In the interests of speed and because of limited disk space, indexes without full positional information were used. This meant that it was not possible to use passage-searching. Apart from this, the runs were done in the same way as the ad hoc, but with parameters intended to maximize precision at 20 documents.

Several pairs of runs were done, but only one—based on the full topic statements—was submitted.

Automatic routing

The emphasis was on term weighting using iterative methods on a number of training databases. City's TREC-5 methods were compared with a type of simulated annealing technique. The latter was very greedy, and tended to result in serious over-fitting. Successive runs using the same parameters would result in very different term-weights, and test results bore no predictable relation to training scores. This effect was lessened by merging a number of queries. Eventually it was decided to use a technique based on City's TREC-5 methods followed by a limited amount of annealing, followed by six–twelve-fold merging of queries. Most of the experiments were done using TREC-5 training and test data. It was possible to improve on City's TREC-5 results but only by a few percent.

*Centre for Interactive Systems Research, Department of Information Science, City University, Northampton Square, London EC1V 0HB, UK. email {sw,ser}@is.city.ac.uk, bougha@irit.fr

†University of Exeter. email gareth@dcs.exeter.ac.uk. Currently Visiting Fellow, Toshiba Corporation, Japan

‡University of Cambridge. email Karen.Sparck-Jones@cl.cam.ac.uk

It was difficult to decide what training data to use for the TREC–6 runs. In the event, both the runs submitted only used the TREC–5 FBIS (filtering) data. One run was based on use of the full (filtering) training data both as term source and for deriving weights; for the other, the training set was split into "odd" and "even" with one half used as term source and the other for weighting. The second of these runs was one of the best submitted.

Filtering track

The filtering work was essentially only a small extension of the routing task effort. The pool of merged routing queries was used, but query selection was based on maximizing (over the training data) each of the utility functions for each topic. Two triples of runs were submitted. Both these sets compared very favourably with other participants' results.

QSDR

Some small-scale experiments were run at Cambridge, using Okapi-type methods, with the QSDR data. These tests gave some indication (albeit qualified by the size of the experiment) that the methods are sufficiently robust to give satisfactory performance with appropriate tuning.

2 Okapi at TRECs 1–5

The search systems City have always used for TREC are descendants of the Okapi systems which were developed at the Polytechnic of Central London[1] between 1982 and 1988 under a number of grants from the British Library Research & Development Department and elsewhere. These early Okapi systems were experimental highly-interactive reference retrieval systems of a probabilistic type, some of which featured automatic query expansion [1, 2, 3].

For TREC–1 [4], the low-level search functions were generalized and split off into a separate library — the Okapi Basic Search System (BSS). User interfaces or batch processing scripts access the BSS using a simple command language-like protocol.

City's TREC–1 results were very poor [4], because the classical Robertson/Sparck Jones weighting model [5] which Okapi systems had always used took no account of document length or within-document term frequency.

During TREC–2 and TREC–3 a considerable number of new term weighting and combination functions were tried; a runtime passage determination and searching package was added to the BSS; and methods of selecting good terms for routing queries were developed [7, 8]. During the TREC–2 work "blind" query expansion (feedback using terms from the top few documents retrieved in a pilot search) was tried for the first time in automatic ad hoc experiments, although we didn't use it in the official runs until TREC–3. Our TREC–3 automatic routing and ad hoc results were both relatively good.

TREC–4 [9] did not see any major developments. Routing term selection methods were further improved.

By TREC–5 many participants were using blind expansion in ad hoc, in some cases more successfully than City [10, 11]. In the routing, we tried to optimize term weights after selecting good terms (as did at least one other participant); our routing results were again among the best, as were the filtering track runs.

3 The system

3.1 The Okapi Basic Search System (BSS)

The BSS, which has been used in all City's TREC experiments, is a set-oriented ranked output system designed primarily for probabilistic–type retrieval of textual material using inverted indexes. There is a family of built-in weighting functions as defined below (equation 1) and described more fully in [8, Section 3]. In addition to weighting and ranking facilities it has the usual boolean and quasi-boolean (positional) operations and a number of non-standard set operations. Indexes are of a fairly conventional inverted type. There were again no major changes to the BSS during TREC–6.

Weighting functions

All TREC–6 searches used varieties of the Okapi **BM25** function first used in TREC–3 (equation 1).

[1] Now the University of Westminster.

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} + k_2.|Q|.\frac{avdl-dl}{avdl+dl} \tag{1}$$

where

Q is a query, containing terms T
$w^{(1)}$ is either the Robertson/Sparck Jones weight [5] of T in Q

$$\log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \tag{2}$$

or else a slightly modified and more general version which takes account of non-relevance as well as relevance information [6]

$$\frac{k_5}{k_5+\sqrt{R}}(k_4+\log\frac{N}{N-n})+\frac{\sqrt{R}}{k_5+\sqrt{R}}\log\frac{r+0.5}{R-r+0.5}-\frac{k_6}{k_6+\sqrt{S}}\log\frac{n}{N-n}-\frac{\sqrt{S}}{k_6+\sqrt{S}}\log\frac{s+0.5}{S-s+0.5} \tag{3}$$

N is the number of items (documents) in the collection
n is the number of documents containing the term
R is the number of documents known to be relevant to a specific topic
r is the number of relevant documents containing the term
S is the number of documents known to be nonrelevant to a specific topic
s is the number of nonrelevant documents containing the term
K is $k_1((1-b)+b.dl/avdl)$
k_1, b, k_2, k_3 and k_4 are parameters which depend on the on the nature of the queries and possibly on the database. For the TREC–6 experiments, k_1 was 1.2 and b was 0.75 except where stated otherwise; k_2 was always zero and k_3 anything from 0 to 1000; when there was not much relevance information -0.7 was a good value for k_4, otherwise zero.
k_5 and k_6 determine, in equation 3, how much weight is given to relevance and non-relevance information respectively. Typical ranges are 0–4 for k_5 and 4–∞ for k_6
tf is the frequency of occurrence of the term within a specific document
qtf is the frequency of the term within the topic from which Q was derived
dl and $avdl$ are the document length and average document length (arbitrary units) resp.

When k_2 is zero, as it was for all the results reported here, equation 1 may be written in the simpler form

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} \tag{4}$$

Nonrelevance information

The extension to the basic weighting formula given by equation 3 is motivated mainly by the desire to make use of explicit judgment of nonrelevance, rather than relying entirely on the "complement" method, by which all documents not known to be relevant are assumed to be nonrelevant. There is a fuller discussion in [6]. This formula might be used in various environments; the particular use reported here has to do with "blind" expansion, where there are no explicit judgments of relevance. Further detail is given below.

Term ranking for selection

In [8] there is a brief discussion of some alternative ways of ranking potential expansion terms. It appeared that no method was superior to the method proposed in [12] by which terms are ranked in decreasing order of $TSV = r.w^{(1)}$. In line with the "nonrelevance" version of $w^{(1)}$ (equation 3) Boughanem has proposed the more general function

$$TSV = (r/R - \alpha s/S).w^{(1)} \tag{5}$$

where $\alpha \in [0,1]$ and r, R, s and S are as above.

Passage determination and searching

Since TREC–3 the BSS has had facilities for search-time identification and weighting of any sub-document consisting of an integral number of consecutive paragraphs. It was described, and some results reported, in [8]. Passage searching almost always increases average precision, by about 2–10 percent, as well as recall and precision at the higher cutoffs. It often, surprisingly, reduces precision at small cutoffs, so is not used in pilot searches for expansion runs. Unless stated, and except for the VLC runs, where the indexes contained insufficient information, and the QSDR runs, passage searching was used in all the runs reported in this paper; comparisons between passage and non-passage runs can be seen in a few of the tables.

3.2 Hardware

There were three dedicated TREC Suns: two 143 MHz single-processor Ultra 1s with an inadequate 128 MB each and an SS20 with 160 MB. Two SS10s were also used, and a third Ultra 1 with 64 MB was borrowed for a time. There was about 60 GB of disk storage, most of it attached to the Ultras and one of the SS10s, and two tape drives. The machines were connected by a local segment of slow ethernet.

3.3 Database and topic processing

For interactive purposes (reported elsewhere) it is necessary to provide for the readable display of documents. Since we have not (yet) implemented a runtime display routine, nor adequate parsing and indexing facilities, for SGML data, all the TREC input text is subjected to batch conversion into a uniform displayable format before further processing. This is done by means of hacked up shell scripts specific to the input dataset. The output records always have three fields: document number, any content unsuitable for indexing (or not to be searched—such as controlled descriptors in some datasets), and the searchable "TEXT" and similar portions.

All the TREC text indexing was of the keyword type. With the exception of the VLC databases a few multiword phrases such as "New York", "friendly fire", "vitamin E" were predefined and there was a pre-indexing facility for the conflation of groups of closely related or synonymous terms like "operations research" and "operational research" or "CIA" and "Central Intelligence Agency". A stemming procedure was applied, modified from [13] and with additional British/American spelling conflation. The stoplist contained 222 words.

Initial topic processing deleted terms such as "document", "describe(s)", "relevan...", "cite..." from any description, narrative or summary fields. What is left was then processed in the same way as text to be indexed. There was a facility for producing adjacent pairs of terms from the topic statements but this was not used during TREC–6. (City have repeatedly experimented with term pairs in previous TRECs, always with negligible benefit.)

4 Automatic ad hoc and VLC track

4.1 "Blind" query expansion

Since TREC–2 City, in common with several other participants, have been experimenting with generating queries for ad hoc searching with the aid of assumed relevant and, more recently, assumed nonrelevant documents, retrieved by means of a pilot search. In general, terms are extracted from the assumed relevant documents and assigned weights using equation 2 or 3. Resulting terms which also occur in the topic statement may have their weights modified by using a positive value of k_3 in the equation, and sometimes all topic terms have their weights increased by a constant factor. Finally, query terms are selected from the resulting pool, usually in descending *TSV* order. There is of course no need to use the target database alone as term source; it is fairly obvious that, under suitable conditions on the content of the database, the larger the collection the greater will be the density of relevant documents.

We have done a very large number of test runs using blind expansion, but these have resulted in very little in the way of guidelines. In our original TREC–2 experiments (using just the target database for the pilot search) we noted that almost any topic would benefit from *some* expansion, even where it turned out that none of the pseudo-relevant documents was really relevant. Unfortunately, there is very wide variation between topics: an expansion techique which is good for one may be bad for another. We have so far found no effective way of choosing expansion methods to suit individual topics.

There is a quite unmanageable number of independent variables: method of derivation of the pilot query, pseudo-relevance and -nonrelevance decisions, extraction and weighting of terms, etc. We choose the parameters of the pilot search (k_1, b, k_3, k_4) to maximize (we hope) precision at cutoff R, where R is the intended number of pseudo-relevant

documents. We then assume that documents 1–R are relevant, skip the next G and assume the following S are non-relevant (having retrieved $R + G + S$ documents). Sometimes, long documents, over 10,000 bytes say, are removed from the R-set[2]. Terms are extracted from each remaining document[3] in the R-set, and n, r and s determined (see Section 3.1). These term records are then run against the topic statement and records for topic terms have their within-topic frequency added. We now have a set of terms, perhaps numbering several thousand, for each topic, from which queries can be constructed.

Many different queries can be constructed from a single term-set. All six[4] of the BM25 parameters (Section 3.1) can be varied. The weight bonus percentage T given to topic terms can be varied, as can the method of term selection. We have tried the following selection methods. In all cases terms are weighted and arranged in descending TSV order.

1. The top t terms are selected.
2. All terms with TSV above a fixed threshold, or above a fixed proportion of the greatest TSV value, are selected.
3. All topic terms are selected together with the best a non-topic terms.
4. If there are nt topic terms, all topic terms are selected together with the best $A\%$ of nt non-topic terms.

Refinements may be added to any of 1–4 above, either to improve results or for efficiency reasons. These include

- exclude terms with very large n
- exclude non-topic terms with small n
- exclude terms with low r
- exclude non-topic terms containing a digit.

4.2 Ad hoc runs

These are summarised in Table 1. All reported runs used passage searching, which gave gains of up to about 12% in average precision. All runs also excuded non-topic terms containing a digit, and in some cases there were rather weak restrictions on r and n.

Three official runs were submitted: city6al (long), city6ad (short) and city6at (title). All used blind expansion from initial searches of a database made up of disks 1–5 and the TREC–4 routing data. The procedures used were based on those which had been most successful in trials with TREC–2, 3 and 5 data. We report also runs using title + description fields.

- The long run used the top 30 terms from the top 15 documents retrieved in the pilot search, documents longer than 10,000 characters being discarded. The terms were weighted using equation 3 with $G = S = 500$, $k_5 = 1$ and $k_6 = 64$. Topic terms had their weights multiplied by 2.5. The terms were ranked by TSV with $\alpha = 0.15$ in equation 5. There was a mistake in the pilot run script, k_3 being treated as zero instead of 7 as intended, so versions of this run have been repeated with two corrected termsets[5].
- The title + description run used the top 30 terms from the top 10 documents retrieved in a pilot search of the disks 1–5 database with pilot $k_3 = 7$ and documents longer than 10,000 characters being discarded. The terms were weighted using equation 3 with $G = S = 500$, $k_5 = 1$ and $k_6 = 64$. $k3 = 1000$ in the final search.
- The official short (description) run used the top 24 terms from the top 10 documents retrieved using $k3 = 7$ and k_4=0, documents longer than 10,000 characters being discarded. The terms were weighted using equation 3 with $G = S = 500$, $k_5 = 1$ and $k_6 = 128$. Topic terms had their weights multiplied by 2.5. The alternative expansion method reported in the table used $k5 = 2$, $k_6 = 64$ on topic (description) terms + an additional 1.75 times the topic length of non-topic terms.
- The title run used topic terms + the top 20 non-topic terms from the top 7 documents retrieved using $k_3 = k_4 = 0$, documents longer than 10,000 characters being discarded. The terms were weighted using equation 3 with $G = S = 500$, $k_5 = 1$ and $k_6 = 128$. Topic terms had their weights multiplied by 3.5.

[2]This does not seem to be beneficial, but does speed the process of query construction.
[3]Sometimes terms are extracted just from the "best" passage in each document, but again we have not found this of noticeable benefit.
[4]Or seven, but k_2 is nearly always zero.
[5]The corrected termsets were derived from a slightly smaller database consisting of disks 1, 2, 3, 4 and 5.

Table 1: Automatic ad hoc results

Method	AveP	P10	P15	P20	RPrec	Rcl
Long topics						
Official city6al, described in 4.2	0.233	0.394	0.357	0.332	0.260	0.525
Corrected version of city6al (pilot $k_3 = 7$)	0.262	0.480	0.435	0.393	0.286	0.581
As previous + final $k_3 = 1000$	0.298	0.508	0.459	0.435	0.310	0.595
As previous but pilot $k_3 = 1000$ and pilot $k_4 = -.7$	0.305	0.522	0.467	0.430	0.322	0.602
No expansion, $k_3 = 1000$, $k_4 = -.7$	0.288	0.480	0.436	0.402	0.320	0.581
Title + description						
Described in 4.2	0.298	0.484	0.443	0.399	0.324	0.582
No expansion, $k_3 = 1000$, $k_4 = -.7$	0.282	0.450	0.405	0.370	0.318	0.544
Description only						
Official city6ad, described in 4.2	0.216	0.356	0.309	0.283	0.250	0.426
Alternative expansion method	0.226	0.354	0.315	0.293	0.257	0.441
No expansion, $k_3 = 1000$, $k_4 = -.7$	0.210	0.320	0.299	0.281	0.249	0.440
Short topics (title only)						
Official city6at, described in 4.2	0.288	0.438	0.393	0.367	0.322	0.555
No expansion, $k_3 = k_4 = 0$	0.251	0.408	0.355	0.336	0.296	0.502

4.3 Very large collection (VLC) track

This was quite a relaxation after the other tracks. There were no preliminary experiments. Any problems were logistical. The Okapi system had never been tried on a database more than a quarter of the size of the new compendium. Two modifications had to be made. Since Sun operating systems (prior to Solaris 2.6) do not allow files to exceed 2 GB Okapi database textfiles may comprise a number of physically separate volumes, the maximum number of volumes had to be increased from 8 to 16 to cater for the 20 GB of text. Secondly, it was clear we would not have enough free disk space, at least not locally, to store the text and an estimated 12 GB of inverted indexes. Hence it was decided to create a new form of index which would not contain full within-document positional information. This brought the index overhead down from about 60% to about 20% of the textfile size. Once these alterations had been done there only remained the operational complications of scheduling the tape-reading, decompression, conversion via Okapi exchange format to runtime format; followed by parsing and term generation (done in five or six portions on several machines), and finally merging of the resulting streamers and the production of a dictionary and single inverted file (in three volumes).

VLC results

These are summarised in Table 2. Note that except for the official runs a considerable number of non-assessed documents were retrieved, so it is likely that "true" results are somewhat better than the ones shown.

The official run, city6vl, (and the baseline run city6vbl), used the top 25 terms by *TSV* (method 1 in Section 4.1) from the top 15 documents retrieved by a pilot search using the full topic statements with $k_4 = 0$, documents longer than 10,000 characters being discarded. Terms were weighted using equation 3 with $k_5 = 1$ and $G = S = 0$, and topic terms had their weights multiplied by 2.5. Final k_3 was zero. There were two mistakes in the scripts which executed this run: the pilot k_3 should have been 7 but was treated as zero, and there was a fault in the term ordering procedure. It was not possible to use passage searching because of the absence of positional information in the indexes. This run came second on precision at cutoff 20 in the official results (ignoring an impressive but presumably manual run from the University of Waterloo).

When the mistakes were discovered a new termset was generated, and gave the results shown under "Corrected city6vl". This gives worse precision at 20, although better on the other statistics. A run using reweighted topic terms only was impressive, and a run with no expansion at all did beter than most of the expanded ones. The best result of all was obtained by using a final k_3 of 1000 and the method of the official city6vl.

In the baseline test, the official run was poor, but a run with no expansion was surprisingly good.

Table 2: VLC results

Method	P5	P10	P15	P20	% unassessed docs
Official city6vl	0.584	0.562	0.532	0.515	0.0
Corrected city6vl	0.592	0.578	0.547	0.506	19.1
As city6vl with final $k_3 = 1000$	0.668	0.614	0.585	0.548	8.8
As corrected city6vl with final $k_3 = 1000$	0.612	0.594	0.564	0.526	14.8
As corrected city6vl, final $k_3 = 1000$, reweighted topic terms only	0.648	0.600	0.556	0.529	14.3
No expansion, $k_3 = 1000$, $k_4 = -.7$	0.672	0.618	0.565	0.517	13.6
As previous but $n < 800000$	0.656	0.582	0.551	0.517	14.4
Official baseline city6vbl	0.404	0.382	0.337	0.320	0.0
Baseline no expansion, $k_3 = 1000$, $k_4 = -.7$	0.520	0.440	0.407	0.374	9.8

VLC timings

The figures in Table 3 are approximate wall-clock times, corrected where necessary to as near as possible what they would have been if all run on one of our Ultra 1s. No correction has been attempted for varying network, disk and CPU loadings from unassociated processes. As expected, most times look roughly linear in data size (the baseline data consisted of a systematically sampled 10% of the main task data). There should be a rather small logarithmic component in the indexing times: this would doubtless show up in the CPU times. It is not altogether clear why expansion term generation took relatively longer for the baseline task, but the explanation may lie in the very inefficient scripts used which have an overhead proportional to the number of topics.

Table 3: VLC processing times in wall-clock hours, main and baseline tasks

Process	Time (hours) main	baseline
Uncompress raw data	3.3	0.3
Strip and convert uncompressed data to Okapi runtime format	28.9	2.4
Parse and index	70.8	7.2
Generate expansion term pool and queries	17.0	2.6
Execute 50 25–term queries	1.0	0.1

4.4 Discussion of ad hoc

City's description and title runs did well, and the corrected long topic runs also compare very favourably with other TREC–6 results. It is quite clear that "blind" query modification is beneficial provided that a large enough database is available, even when use of the pseudo-relevant documents is limited to reweighting the original query terms. However, most of the more successful participants are using something similar. Passage searching almost always increases average precision and recall, and the new weighting formula equation 3 is undoubtedly of some benefit [6], although this is not demonstrated in this paper. The new formula has two advantages: smallish non-zero values of k_5 mean that limited use can be made of little, or dubious, positive relevance information; and it appears that quite large k_6 values enable some use to be made of negative information.

5 Automatic routing and filtering

5.1 Routing

Training sets

To start with, all the (positive) relevance judgments for the TREC–6 routing topics were used. It is probably inadvisable to use datasets with incomplete relevance judgments for routing term selection, so about six databases had to be used, involving a lot of time, both human and machine. For most topics there were a large number of relevant documents (mean 579, median 510). After a number of trials had been done using the full set of judgments

it was decided to try using the filtering training set (TREC-5 routing database) only, although the judgments were said to be incomplete for nine of the 47 topics. For this set the mean was 123 (median 45).

Outline

Terms were extracted from relevant documents for some database and weighted using the customary Okapi formula $w^{(1)}$ (equation 2; the new "nonrelevance" formula equation 3 was not used in the routing). The terms were then arranged in descending TSV order. A fixed number of terms were taken from the top of the term list and subjected to selection and/or reweighting procedures. Sets were formed and scored in some way using either the same or a different database and set of judgments with the object of obtaining as high a score as possible. The only scoring function used for TREC–6 was non-interpolated average precision at cutoff 1000. Finally, a number of the resulting query sets were merged.

Term weight optimization

For TREC–5 we tried some reweighting experiments, which appeared to give a small improvement over simple selection of terms. Since we had acquired two extra, and faster, computers this year it was decided to try some more drastic reweighting methods. It seemed possible that some type of simulated annealing might work, although it might be thought that practical scoring functions would not have the relative "smoothness" for this to work satisfactorily. A simple but very compute-intensive procedure was developed and gave encouraging scores during reweighting. However, when applied predictively results varied rather wildly. It appeared that the procedure was giving serious overfitting to the selection database. The next step was to combine the new procedure with the deterministic reweighting which we had used for TREC-5. Eventually we tried two to four passes of deterministic single-term reweighting followed by a rather mild annealing process. The latter almost always gave a noticeable increase in selection score but it is not yet clear whether there was any real gain when applied predictively to the test set.

The simulated annealing procedure

This proceeds in a number of stages, the "temperature" being reduced between each one, with a final "quench" at zero temperature. At all times two configurations are saved: a local "best" and a global best. Each stage consists of a number of iterations in each of which the weights of randomly selected terms are increased or decreased and a new score calculated. If the new score is higher than the local best the new configuration is retained and becomes the current best. If it is lower than the current best it is nevertheless retained (as current best) with a probability which decreases with T, the current temperature[6]. The motivation for sometimes keeping an apparently worse configuration is that this may enable escape from a local maximum. The hope is that a lot of local maxima may be explored and one ends up with the best one, or at least a rather good one. Finally, if the local best from the "quench" stage is worse than the global best the latter becomes the final result.

Obviously, there are a considerable number of tuning parameters: for example the distributions of the number of terms to have their weights altered and the extent of weight variation, the temperature reduction function, rules for ending stages and for ending the whole procedure. We were not able to explore the possibilities anywhere near exhaustively, and results so far are not particularly encouraging.

Merging runs

As in previous TRECs, a number of term sets from different selection procedures were merged to form the final query, thus reducing over-fitting. Where partitioned databases were used for training all runs were duplicated: terms from "odd" (say) and optimization on "even" followed by the reverse; the two term sets would then be merged. The simulated annealing led to quite wild variation in the magnitude and range of weights, so a term set to be merged with another would have its weights normalised relative to the median weight of the set.

5.2 Automatic routing results

These are given in Table 4.

[6]The usual rule, which we used, is "accept worse score with probability $\exp(-(best\_score - new\_score)/T)$".

As mentioned above, both sets of submitted queries (city6r1 and city6r2) were derived using only the TREC–5 routing database and the TREC–6 filtering training relevance judgments. Had time allowed we should probably have used some additional pre–TREC–5 training information for some of the topics with few known relevant FBIS documents. The difference between the two sets is that for city6r1 the terms came from one half of the database and the optimization was done on the other half; whereas for city6r2 the whole database was used both as term source and for optimization. Experiments with TREC–5 routing data had suggested that the former method was likely to give slightly better results, although the relatively small number of TREC–6 training judgments made it dangerous to assume that this would still hold. Hence it is quite surprising that the cityr1 result turned out so much the better of the two. However, there were other differences between them. Both sets of queries were formed by merging a number of query sets, but 24 (12 pairs) were used to form city6r1 and only six for city6r2; city6r1 used four deterministic weight variation passes with just a final "quench" stage; city6r2 had three deterministic passes followed by four stages of simulated annealing.

All the optimization runs started with a term pool of size 100 (in previous TRECs we had found a small gain from using up to 300 terms, but the simulated annealing would have been too slow on sets of this size). City6r1 ended with a mean of 138 terms per query and city6r2 with 86, but many terms had very low weights and the queries could probably be reduced by 25 percent or more without greatly affecting results.

Table 4: Automatic routing results

Run	AveP	P5	P10	P15	P20	P30	RPrec	Rcl
city6r1	0.408	0.698	0.653	0.606	0.578	0.548	0.411	0.809
As city6r1 but without passages	0.399	0.706	0.651	0.604	0.581	0.545	0.409	0.802
city6r2	0.378	0.668	0.626	0.590	0.554	0.523	0.399	0.760
As city6r2 but without passages	0.368	0.672	0.619	0.580	0.555	0.515	0.392	0.753

5.3 Filtering

Filtering estimation procedure

A pool of queries was selected from the merged queries produced during the routing training. Each query was executed against a training database to produce a standard TREC output file with the addition of a field for each document containing the relevance assessment (relevant, nonrelevant or not assessed). These output files were run through a script which calculated the value of each of the utility functions at each rank. The threshold weight which maximized each function was then found. Output from this stage was of the form

⟨topic⟩⟨func⟩⟨value⟩⟨thresh⟩⟨rank⟩⟨prec⟩⟨recall⟩⟨# rels⟩⟨query file⟩

The lines which gave the highest value for each function for each topic were extracted from this, thus giving the queries and thresholds to be used.

Filtering results

Three triples of queries and thresholds were submitted, city6f1[1-3] and city6f2[1-3]. City6f1 had what ought to have been a sounder basis; it was produced by executing routing queries—whose weights had been derived using one half of the training database—against the other half of the database. City6f2 used queries where both halves of the database had contributed to the weights, run retrospectively against the whole database. The latter must have led to a substantial overestimation of maximum function values, though it is not obvious what effect this would have on the estimation of thresholds. From the comparisons with median scores (Table 5) it looks as if city6f2 may in fact have been slightly the better of the two.

5.4 Discussion of routing results

One of the most important aspects is avoidance of over-fitting of queries to the training database. Given enough computer time it is not difficult to obtain very good retrospective scores, but the resulting queries produce poor results when applied predictively. We have tried various appealing ideas along the lines of rejecting rare terms or only taking "good" passages from long relevant documents, but have always found these to be if anything marginally

Table 5: Filtering results (pooled evaluation): comparison with median scores

Run	function	$\geq$ med.	best	$<$ med.	worst
city6f11	F1	36	8	11	0
city6f21	F1	37	4	10	0
city6f12	F2	37	5	10	0
city6f22	F2	41	5	6	0
city6f13	ASP	45	5	2	0
city6f23	ASP	45	8	2	0

detrimental. In practice, it seems to be best to merge queries from as many reasonably promising sets as possible. It may be that over-fitting was less of a problem in TREC–6, where for the first time the test database was (presumably) rather "similar" to the (filtering) training database, a more realistic setup than in previous TREC routing tracks.

From the point of view of real life routing situations, there is an urgent need to work on techniques for properly using relevance information as it increases from an initial zero. Our results are not good on topics for which there are few relevant documents. Blind expansion is no good unless there is a large database of documents of a suitable type. In particular some experiments should be done on query optimization with little relevance information. During our TREC–5 ad hoc work we did some runs where queries were "optimized" on pseudo-relevant documents, with surprisingly good results.

6 QSDR experiments for City University

We were unable at Cambridge, for independent reasons, to do the actual SDR speech test. We therefore carried out only some rather simple, lightweight tests using the baseline SRT data. But these were of the kind drawing on IR experience that the QSDR option was intended to encourage. Thus we treated QSDR as a way of checking out the Okapi-type retrieval methods, already applied to speech data in the Cambridge VMR project [19], with (a) new, different, document data (b) known-item searching rather than conventional queries. We set aside questions of whether the Cambridge speech recognition system could do any better than the one used for the baseline (without or with tuning to the application), and also any retrieval strategies tied to speech (eg various fusion ones combining different recognition strategies).

Thus for both the baseline SRT and the LTT versions of the documents we compared search performance for unweighted terms (UW), terms with only collection frequency weighting (CFW), and terms with full Okapi-style combined weighting (CW, aka BM25, see equation 1 or [20]). We were interested in whether relative performance for these strategies was the same for the SRT and LTT data, and how the weighting formulae in particular behaved in (a) improving targeting on the known item and (b) compensating for speech recognition deficiencies.

However the small data scale, and especially tiny training query sample, made adaptation to the TREC case uncertain, and the results given below must be taken with much salt. The small document set size also limits inference for performance with the larger test query set. Using the training data we applied, as usual, stop listing (with the standard van Rijsbergen stop list) and stemming (Porter), and after experiment set the CW tuning constants b and K. Some method for dealing with acronyms is needed, but without any in the training query set we were not able to choose a suitable one.

Performance for the training set (though only indicative, with so few queries) suggested that respectable performance for SRT against LTT, and decent performance for the known-item type searching, should be obtained for the test data, as well as the predicted best results for CW. This was borne out in practice, as shown below. In particular, there are clear performance gains for CW with the SRT test data, and though the expected run length is still much worse for SRT than LTT, CW again gives the best results.

Table 6 illustrates CW performance with parameter settings the same for the LTT and SRT data. The test data runs labelled cw2, with the parameter settings based on the training data, are the officially submitted runs 'citysdrR2' and 'citysdrB2', for LTT and SRT respectively. Setting the parameters differently for LTT and SRT, as in the runs citysdrR1 and citysdrB1, makes little difference, as predictable for such small data. On the other hand, as the cw4 figures in Table 6 show, setting the CW parameters to suit the test data rather than applying training data ones as in the official runs, could be expected to improve performance; and of course such document-file linking for weighting formulae for ad hoc retrieval is wholly feasible.

Performance for the official runs, taking citysdrR2 and city sdrB2 as representative and using percent queries retrieving the known item at rank 1, was above the median, in the leading cohort with roughly similar performance.

This was reassuring. However, while our tests could be viewed as checking the relative values of weighting formulae, and consistency under scaling up, what we were really testing was the viability of a completely routine approach to spoken document retrieval without any adaptation, in either speech processing or retrieval mechanisms, to the spoken document data and the particular retrieval task. Thus IBM's recogniser was deliberately not tuned to the data, but run blind, to produce the SRT transcripts; and we did not attempt to tune the retrieval system to known-item searching, e.g. by adopting a precision-oriented strategy. Our results therefore suggest that given a good recogniser, like IBM's, and sound retrieval methods, respectable performance can be obtained, though it is possible that additional strategies e.g. data fusion on the speech side or query expansion on the retrieval side, could be of use. However the inference just drawn about competitive performance must be heavily qualified because the retrieval test was so tiny, and the task defined by the requirement to retrieve the specified known items seems to have been rather easy.

Table 6: QSDR results

		Training data			Test data			
		uw	cfw	cw2[a]	uw	cfw	cw2[a]	cw4[a]
Ave Prec	LTT	0.70	0.69	0.78	0.55	0.74	0.81	0.84
	SRT	0.57	0.65	0.65	0.46	0.63	0.68	0.72
Exp Run Lngth	LTT	5.67	7.33	1.50	13.02	5.98	5.41	5.06
	SRT	12.17	9.17	5.00	33.27	16.51	14.29	12.84
Mean Recip	LTT	0.70	0.69	0.81	0.55	0.74	0.81	0.84
	SRT	0.51	0.68	0.73	0.44	0.61	0.68	0.72

[a]setting constants b and K the same for LTT and SRT
cw2 for training data: $b = 0.25$, $K = 2.5$
cw2 for test data uses the training settings
cw4 uses settings chosen for the test data: $b = 0.5$, $K = 1.0$

References

[1] Mitev, N.N., Venner, G.M. and Walker, S. *Designing an online public access catalogue: Okapi, a catalogue on a local area network.* British Library, 1985. (Library and Information Research Report 39.)

[2] Walker, S. and Jones, R.M. *Improving subject retrieval in online catalogues: 1. Stemming, automatic spelling correction and cross-reference tables.* British Library, 1987. (British Library Research Paper 24.)

[3] Walker,S. and De Vere, R. *Improving subject retrieval in online catalogues: 2. Relevance feedback and query expansion.* British Library, 1990. (British Library Research Paper 72.) ISBN 0-7123-3219-7

[4] Robertson, S.E. *et al.* Okapi at TREC. In: [15], p21–30.

[5] Robertson, S.E. and Sparck Jones K. Relevance weighting of search terms. *Journal of the American Society for Information Science* 27, May–June 1976, p129–146.

[6] Robertson, S.E. and Walker S. On relevance weights with little relevance information. In *Proceedings of the 20th annual international ACM SIGIR conference on research and development in information retrieval.* Edited by Nicholas J Belkin, A Desai Narasimhalu and Peter Willett. ACM Press, 1997. p16–24.

[7] Robertson, S.E. *et al.* Okapi at TREC–2. In: [16], p21–34.

[8] Robertson, S.E. *et al.* Okapi at TREC–3. In: [17], p109–126.

[9] Robertson, S.E. *et al.* Okapi at TREC–4. In: [18], p73–96.

[10] Beaulieu, M.M. *et al.* Okapi at TREC–5. In: [11].

[11] *The Fifth Text REtrieval Conference (TREC–5).* Edited by D.K. Harman. Gaithersburg, MD: NIST, 1997.

[12] Robertson, S.E. On term selection for query expansion. *Journal of Documentation* 46, Dec 1990, p359–364.

[13] Porter, M.F. An algorithm for suffix stripping. *Program* 14 (3), Jul 1980, p130-137.

[14] Robertson, S.E., Walker, S. and Hancock-Beaulieu, M.M. Large test collection experiments on an operational, interactive system: OKAPI at TREC. *Information Processing & Management* 31 (3), p345–360, 1995.

[15] *The First Text REtrieval Conference (TREC–1).* Edited by D.K. Harman. Gaithersburg, MD: NIST, 1993.

[16] *The Second Text REtrieval Conference (TREC–2).* Edited by D.K. Harman. Gaithersburg, MD: NIST, 1994.

[17] *Overview of the Third Text REtrieval Conference (TREC–3).* Edited by D.K. Harman. Gaithersburg, MD: NIST, 1995.

[18] *The Fourth Text REtrieval Conference (TREC–4).* Edited by D.K. Harman. Gaithersburg, MD: NIST, 1996.

[19] Jones, G.J.F., Foote, J.T, Sparck Jones, K. and Young, S.J. *Video Mail Retrieval using voice: Report on topic spotting,* Technical Report 430, Computer Laboratory, University of Cambridge, 1997.

[20] Robertson, S.E. and Sparck Jones, K. *Simple, proven approaches to text retrieval,* Technical Report 356, Computer Laboratory, University of Cambridge, updated May 1997.

Okapi Chinese text retrieval experiments at TREC–6

Xiangji Huang* S E Robertson*

1 Introduction

The focus of the Okapi TREC–6 Chinese experiments is on investigating the effectiveness of different automatic indexing methods and phrase weighting for retrieval based on probabilistic models over Chinese text. We compare different probabilistic weighting methods based on a range of word and single character approaches.

There are two indexing methods used in our experiments. One indexing method is to use linguistic units (words, compound words and phrases) in texts as indexing terms to represent the texts. We refer to this method as the word approach. For this approach, text segmentation, which divides text into linguistic units, is regarded not only as a necessary precursor but also as a bottleneck of this kind of system [1]. The other method for indexing texts is based on single Chinese characters, in which texts are indexed by the characters appearing in the texts [2]. By using single character approaches, a search could be conducted for any multi-character word or phrase identified at search time, no matter whether this word or phrase is in the dictionary.

Three automatic runs city97c1, city97c2 and city97c3 were submitted in TREC–6. All the three runs were based on the whole topic. City97c1 and city97c3 are for word indexing approach with different parameter values and city97c2 is for character indexing approach.

The runs reported here are all on the TREC–6 collection of 26 new Chinese topics and 164768 documents. The Chinese dictionary we use for our word approach retrieval system contains about 70,000 Chinese words and phrases. Most of these words and phrases come from a manually constructed dictionary in China. We expanded this dictionary while working on the Chinese TREC experiments.

2 Chinese text segmentation

2.1 Concepts and methods

A Chinese word is the minimal linguistic unit that can be used independently. Most modern Chinese words consist of more than one ideographic character and the number of characters in a word varies. Since a Chinese text is a linear sequence of non-spaced or equally spaced ideographic characters, we must either apply a dictionary-based Chinese word segmentation method to the text, or index and search in terms of single Chinese character.

There are different requirements on Chinese text segmentation for different applications. For example, machine translation and natural language processing require the correct segmentation of Chinese text according to Chinese syntax. For the purpose of information retrieval, we may not have to segment Chinese text correctly.

Segmentation based on a big dictionary can be classified into three groups. The first is the longest match, for which text is sequentially scanned to match the dictionary. The longest matched strings are taken as indexing and search tokens and shorter tokens within it are discarded. Since longer tokens in the dictionary are more specific, longest match will generate fewer tokens with more specific meaning.

The second is the shortest match, for which text is sequentially scanned to match the dictionary. The first matched tokens are taken and the match process started from the next character. With the shortest match method, the segmentation process will generate more tokens with less specific meaning. The third is the overlap match, for which tokens generated from the text can overlap each other across the matching boundary.

For the word approach retrieval system in our experiments, we used the longest match algorithm to segment Chinese texts. By applying this algorithm to Chinese TREC collections, approximately 43.6 million words and phrases were identified. Since no dictionary is expected to include all the words, there must be some unmatched

*Centre for Interactive Systems Research, Department of Information Science, City University, Northampton Square, London EC1V 0HB, UK. email {xjh,ser}@is.city.ac.uk

strings left with dictionary based segmentation algorithms. Unmatched strings can be used as tokens for indexing and search. They can also be used to expand the dictionary. In our experiments, we take each character in the string as a token.

For the single character approach, it is a purely mechanical procedure to segment TREC Chinese texts into single characters. An inverted file of about one gigabyte is generated for the character approach retrieval systems in our experiments.

2.2 Query processing

There are two kinds of automatic methods for query processing in our Chinese text retrieval system. One is character-based method, which uses characters, character pairs and multi-character adjacencies as retrieval keywords. Character pairs and multi-character adjacencies are similar to the bigrams and n-grams investigated by some other researchers [3, 4]. The other is word-based method. Given a word segmentation system, similar methods for characters can be applied to words as a way of allowing phrases to contribute to the matching. We refer to this method as the word-based query processing. Table 1 shows nine methods which have been implemented in our systems.

Algorithms	Query processing	Database	Number of weighting methods
M_0	characters	character	1
M_1	characters+character-pairs	character	8
M_2	characters+multi-character adjacencies	character	8
M_3	words	character	8
M_4	words	word	1
M_5	words+word-pairs	character	8
M_6	words+word-pairs	word	4
M_7	words+multi-word adjacencies	character	8
M_8	words+multi-word adjacencies	word	4

Table 1: Illustration of retrieval algorithms

We use M_5 and M_6 for all the TREC experiments reported here. Thus the query-processing is word-based, but both forms of document processing are used. The text in all parts of the Chinese topics were treated in the same way. Text segmentation is applied to the topics first. Only pairs of the adjacent segmented terms which occur in the same subfield of the topic are regarded as new potential phrases. All these segmented terms and new potential phrases are ranked by the values of their weights multiplied by the within-query frequencies. The top 19 terms are used as keywords for searching the word index and for searching the character index.

3 Probabilistic model

3.1 Basic weighting function

The basic weighting functions are based on the Robertson-Sparck Jones weight [5], which approximates to inverse collection frequency (ICF) shown in equation 1 when there is no relevance information.

$$ICF = \log \frac{N - n + 0.5}{n + 0.5} \tag{1}$$

The weighting function we use for Okapi TREC–6 Chinese experiments is given as follows. This function is extended from the BM25 function [7].

$$w = \frac{(k_1 + 1)tf}{K + tf} \log \frac{N - n + 0.5}{n + 0.5} . \frac{(k_3 + 1)qtf}{k_3 + qtf} \oplus k_d y \tag{2}$$

where

N is the number of indexed documents in the collection,
n is the number of documents containing a specific term,
R is the number of known relevant documents for a specific topic,
r is the number of known relevant documents containing a specific term,
tf is within-document term frequency,
qtf is within-query term frequency,

$$y = \begin{cases} \ln(\frac{dl}{avdl}) + \ln(x_1) & \text{if } 0 < dl \leq rel\_avdl; \\ (\ln(\frac{rel\_avdl}{avdl}) + \ln(x_1))(1 - \frac{dl - rel\_avdl}{x_2 . avdl - rel\_avdl}) & \text{if } rel\_avdl < dl < \infty. \end{cases}$$

where dl is the length of the document, $avdl$ is the average document length rel_avdl is the average relevant document length in TREC–5, x_1 and x_2 are two parameters to be set,

the k_i are tuning constants, which depend on the database and possibly on the nature of the topics and are empirically determined,[1]
K equals $k_1((1-b) + b.dl/avdl)$, and
the $\oplus$ in the formula indicates that the following component is added only once per document, rather than for each term.

This formula now defines the weight of a term (that is, the contribution of that term to the total score for the document) in the context of an individual document.

The y bit in BM25 as defined for TREC–5 equals $|Q|.\frac{avdl-dl}{avdl+dl}$, where Q is a query, containing terms T (although it was not actually used in TREC–5, as k_d was set to zero). That means y will decrease with dl, from a maximum as $dl \to 0$, through zero when $dl = avdl$, and to a minimum as $dl \to \infty$. But it should be better that y will reach a maximum as $dl \to rel\_avdl$, through zero when $dl = avdl/x_1$ (or $dl = x_2.avdl$), and to a minimum as $dl \to 0$ (or $dl \to \infty$). So a new y bit was designed for TREC–6, where x_1 and x_2 were set to 3 and 26 respectively, after some experiments on the TREC–5 collection. We may get better results by setting other values for x_1 and x_2.

3.2 Phrase weighting for Chinese

We need weighting functions which will enable us to cope with phrases in a word-based system, and with words or phrases in a character-based system. Suppose, then, that we have a sequence of j adjacent units $t_1 t_2 ... t_j$ (characters or words) constituting a single larger unit (word or phrase). In the Robertson/Sparck Jones model, each unit (large or small) has a "natural" weight, given by the formula; let these be w_{t_i} and $w_{t_1 t_2 ... t_j}$ respectively. Then we can suggest a number of weighting functions which satisfy (or will probably satisfy) the above condition or something like it. Table 2 gives a few such functions which have been implemented.

Weight methods	$w(t_1\ adj\ t_2\ adj \cdots adj\ t_j)$	$w(t_1 \wedge t_2 \wedge \cdots \wedge t_j)$
$Weight_0$	$\sum_{i=1}^{j} w_{t_i} + j^k * w_{t_1 t_2 ... t_j}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_1$	$w_{t_1 t_2 ... t_j}$	$\sum_{i=1}^{j} w_{t_i} - j^k$
$Weight_2$	$w_{t_1 t_2 ... t_j}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_3$	$\sum_{i=1}^{j} w_{t_i} + log\frac{\#(t_1 \wedge t_2 \wedge \cdots \wedge t_j)}{\#(t_1\ adj\ t_2\ adj \cdots adj\ t_j)}$	$\sum_{i=1}^{j} w_{t_i}$
where $\#(t)$ indicates the number of documents containing the term t and k is another tuning constant		

Table 2: Weight methods

None of these functions has a very strong probabilistic basis, beyond the attempt to satisfy the condition. Each has two versions, one applied to words and the other to characters, so there are eight functions in all. The "natural" weights come from equation 2: w_{t_i} is obtained by applying the equation to term t_i, and $w_{t_1 t_2 ... t_j}$ comes from applying the same equation to the combined term $t_1 t_2 ... t_j$.

In our experiments, the value of k was 0.5. The average document length is 891 bytes, The average relevant document length in TREC–5 is 1399 bytes.

[1]For our experiments, the k_i values will be given in Section 5

4 Experimental results and performance comparison

Extensive experiments have been done on an SGI Challenging L machine to investigate: 1. the effect of probabilistic approach on Chinese text retrieval; 2. the difference between word approach and character approach; 3. the effect of different phrase weighting functions and of varying their parameters.

All the results reported here are from Chinese ad hoc experiments on the various TREC collections. These experiments represent only a small proportion of the range suggested by the foregoing discussion. Experiments are still continuing. In our experiments, the relevance judgements for each topic come from the human assessors of NIST. Statistical evaluation was done by means of the latest version TREC evaluation program, for which we are grateful to Chris Buckley. The abbreviated captions in the statistical tables have the following meanings: Average Precision: average precision over all 11 recall points (0.0, 0.1, 0.2,..., 1.0); R Precision: precision after the number of documents retrieved is equal to the number of known relevant documents for a query; Precision 100 docs: precision after 100 documents have been retrieved.

Three automatic runs city97c1, city97c2 and city97c3 were submitted for evaluation in TREC–6. All the three runs were based on the whole topic. City97c1 and city97c3 are for word indexing approach with different parameter values and city97c2 is for character indexing approach. For these three submitted runs, the values of k_1 and k_3 were set to 2.0 and 5.0. For city97c1 and city97c3, the k_d were set to 6 and 15 respectively. The k_d was set to 6 for city97c2.

The three submitted Chinese results for TREC ad hoc experiments are shown in Table 3 and Table 4. All of these three runs for the whole 26 topics are around the median score of the groups participating in the Chinese track.

The results for the different weighting methods of the character approach are presented in Table 5. Table 6 gives some more evaluation results for the word approach. The values of k_1, k_3 and b for the above runs are 2.0, 5.0 and 0.75 respectively. In these two tables, the method of $Weight_1$ gives the best result comparing to the other methods for both the word and character approaches.

Table 7 gives more details of the evaluations of the four runs for the word and character approaches by setting the values of parameter $k_d = 0$ and 15 respectively. From this table, we can see that the character approach city97c02 gives almost identical performance to the word approach city97w02. By setting k_d to 15, the word and character approach perform 7.20% and 11.28% better than the word approach city97w02 with k_d equal to 0.

Run	> median	= median	< median
city97c1	7	0	19
city97c2	10	1	15
city97c3	8	0	18

Table 3: Comparative Chinese Results

Run	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
city97c1	0.4838	2495	0.5119	0.4804
city97c2	0.5047	2589	0.5221	0.4900
city97c3	0.4943	2461	0.5178	0.4835

Table 4: Chinese Ad hoc Results

5 Discussion

Our overall results are a little disappointing. However, we have a great deal more testing to do, which we hope will enable us to make up some of the ground.

One result of particular interest to us relates to the relative success of Weight1. Although, as stated earlier, the formula does not have a strong justification in terms of the probabilistic model, there is at least a qualitative argument which might explain the result. There are two parts to this argument.

It may be seen from Table 2 that Weight1 does not assign the usual sum of individual term weights to the conjunction of a set of terms. Instead it reduces that sum by a certain amount. If we consider the occurrence of a

Run	Weighting Method	k_d	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
city97c01	$Weight_0$	0	0.4563	2525	0.4618	0.4565
city97c29	$Weight_1$	0	0.4808	2565	0.4972	0.4738
city97c03	$Weight_2$	0	0.4730	2519	0.4972	0.4669
city97c04	$Weight_3$	0	0.4223	2423	0.4335	0.4319
city97c05	$Weight_0$	2	0.4596	2532	0.4650	0.4600
city97c30	$Weight_1$	2	0.4900	2579	0.5111	0.4812
city97c07	$Weight_2$	2	0.4832	2541	0.5096	0.4750
city97c08	$Weight_3$	2	0.4271	2424	0.4397	0.4385
city97c09	$Weight_0$	6	0.4632	2538	0.4697	0.4627
city97c32	$Weight_1$	6	0.5047	**2589**	0.5221	0.4900
city97c11	$Weight_2$	6	0.4964	2547	0.5178	0.4819
city97c12	$Weight_3$	6	0.4324	2436	0.4523	0.4819
city97c13	$Weight_0$	8	0.4649	2544	0.4718	0.4638
city97c33	$Weight_1$	8	0.5084	2588	0.5244	0.4927
city97c15	$Weight_2$	8	0.5011	2542	0.5198	0.4854
city97c16	$Weight_3$	8	0.4343	2436	0.4526	0.4504
city97c17	$Weight_0$	10	0.4666	2547	0.4756	0.4638
city97c34	$Weight_1$	10	0.5113	2580	**0.5288**	0.4958
city97c19	$Weight_2$	10	0.5034	2536	0.5205	0.4865
city97c20	$Weight_3$	10	0.4358	2426	0.4551	0.4485
city97c21	$Weight_0$	15	0.4697	2549	0.4787	0.4658
city97c35	$Weight_1$	15	**0.5129**	2560	0.5274	**0.4981**
city97c23	$Weight_2$	15	0.5068	2522	0.5209	0.4888
city97c24	$Weight_3$	15	0.4374	2413	0.4553	0.4488
city97c25	$Weight_0$	20	0.4718	2543	0.4816	0.4662
city97c36	$Weight_1$	20	0.5095	2526	0.5227	0.4950
city97c27	$Weight_2$	20	0.5052	2498	0.5209	0.4888
city97c28	$Weight_3$	20	0.4360	2395	0.4551	0.4504

Table 5: Phrase weighting for character approaches

phrase as a subset of the occurrence of the conjunction of the corresponding terms, and if we give some extra weight to the phrase, the probabilistic model suggests that we should also reduce the scores of the remaining documents in the conjunction (those that do not include the phrase). This in effect is achieved by Weight1.

This result suggests a new look at phrase weighting schemes in English as well as in Chinese.

Acknowledgments

This research is supported by ORS award from Committee of Vice-Chancellors and Principals of United Kingdom and Centenary Scholarship from City University.

References

[1] Wu, Z.; Tseng, G. Chinese text segmentation for text retrieval: Achievement and Problems. *Journal of the American Society for Information Science* 44 (9): 532-541; 1993

[2] Chen, G. On single Chinese character retrieval system. *Journal of Information* 11 (1): p11-18; 1992 (in Chinese)

[3] Chien, L.F. Fast and quasi-natural language search for gigabits of Chinese texts. In *SIGIR 95*: p112-120; 1995.

[4] Willett, P. Document retrieval experiments using indexing vocabularies of varying size. II. Hashing, truncation, digram and trigram encoding of index terms. *Journal of Documentation* 35 (4): p296-305; 1979

[5] Robertson, S.E.; Sparck Jones, K. Relevance weighting of search terms. *Journal of the American Society for Information Science* 27: p129-146; 1976

Run	Weighting Method	k_d	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
city97w01	$Weight_0$	0	0.4195	2419	0.4341	0.4188
city97w02	$Weight_1$	0	0.4609	2434	0.4852	0.4542
city97w03	$Weight_2$	0	0.4586	2425	0.4832	0.4508
city97w04	$Weight_3$	0	0.4561	2447	0.4746	0.4550
city97w05	$Weight_0$	2	0.4251	2433	0.4380	0.4262
city97w06	$Weight_1$	2	0.4704	2469	0.4946	0.4688
city97w07	$Weight_2$	2	0.4683	2464	0.4929	0.4669
city97w08	$Weight_3$	2	0.4630	2476	0.4802	0.4627
city97w09	$Weight_0$	5	0.4321	2436	0.4472	0.4373
city97w10	$Weight_1$	5	0.4810	2490	0.5062	0.4773
city97w11	$Weight_2$	5	0.4792	2485	0.5035	0.4762
city97w12	$Weight_3$	5	0.4702	2493	0.4844	0.4669
city97w13	$Weight_0$	6	0.4340	2440	0.4484	0.4388
city97w14	$Weight_1$	6	0.4838	**2495**	0.5119	0.4804
city97w15	$Weight_2$	6	0.4823	2492	0.5062	0.4800
city97w16	$Weight_3$	6	0.4714	2491	0.4848	0.4681
city97w17	$Weight_0$	8	0.4371	2436	0.4546	0.4462
city97w18	$Weight_1$	8	0.4879	2489	0.5142	0.4812
city97w19	$Weight_2$	8	0.4868	2489	0.5114	0.4800
city97w20	$Weight_3$	8	0.4737	2486	0.4901	0.4696
city97w21	$Weight_0$	10	0.4395	2432	0.4597	0.4477
city97w22	$Weight_1$	10	0.4904	2478	0.5166	0.4812
city97w23	$Weight_2$	10	0.4897	2478	0.5166	0.4800
city97w24	$Weight_3$	10	0.4756	2478	0.4920	0.4712
city97w25	$Weight_0$	15	0.4434	2422	0.4667	0.4492
city97w26	$Weight_1$	15	**0.4943**	2461	**0.5178**	**0.4835**
city97w27	$Weight_2$	15	0.4935	2465	0.5143	0.4819
city97w28	$Weight_3$	15	0.4772	2458	0.4915	0.4723
city97w29	$Weight_0$	20	0.4435	2411	0.4690	0.4535
city97w30	$Weight_1$	20	0.4927	2440	0.5107	0.4808
city97w31	$Weight_2$	20	0.4919	2437	0.5099	0.4808
city97w32	$Weight_3$	20	0.4730	2427	0.4892	0.4731
city97w33	$Weight_0$	50	0.4309	2245	0.4661	0.4446
city97w34	$Weight_1$	50	0.4414	2201	0.4789	0.4581
city97w35	$Weight_2$	50	0.4397	2193	0.4791	0.4565
city97w36	$Weight_3$	50	0.4339	2212	0.4671	0.4485

Table 6: Phrase weighting for word approaches

Run	k_d	Indexing Method	Average Precision
city97w02	0	*word*	0.4609
city97c29	0	*character*	0.4808 (+4.32%)
city97w26	15	*word*	0.4943 (+7.24%)
city97c35	15	*character*	0.5129 (+11.28%)

Table 7: Chinese Ad hoc Results Comparison

[6] Beaulieu, M.M., Gatford, M, Huang, X., Robertson, S.E., Walker, S. and Williams, P. Okapi at TREC–5". In D.K.Harman, editor, *Proceedings of the Fifth Text REtrieval Conference (TREC–5)*. NIST Special Publication 1997.

Interactive Okapi at TREC–6

M M Beaulieu* M J Gatford*

January 1998

1 Experimental setting

A full description of the experimental system and conditions is given in Appendices A and B. Searchers filled in three types of questionnaires. The pre-session questionnaire established the user's experience and profile. In the post search questionnaires, searchers were asked questions regarding the topic, the search and the system used after undertaking each individual search. Finally in the post-session questionnaire, subjects were asked to provide an overview of the experiment. In addition to the questionnaires, searchers noted on a worksheet the different aspects of the topics they encountered whilst they undertook each search.

A total of eight subjects completed forty eight searches, that is three searches on each of the two systems, Okapi and ZPrise. The sessions were divided into two rounds of four searchers. Of the two groups who carried out the twenty-four searches on Okapi, Group A used the same interface as in TREC-5, but with incremental query expansion modified (Appendix A.3.2), and Group B searched a slightly different version which allowed the searcher to cancel the relevance feedback process or clear the query (Appendix A.4).

1.1 Searcher profiles and experience.

Subjects were post-graduate students at Masters level recruited from the Information Science Department at City. As such they had all undertaken some courses in information retrieval and at least in theory had been introduced to probabilistic methods and the principles of relevance feedback and query expansion. However in practice only three out of the eight declared they had some experience in searching ranked output IR systems and only two in systems with relevance feedback. Of those one subject had previously used another version of the Okapi system.

In terms of their overall online experience, half had had a year or less and half had between two and four years experience in searching systems with mouse-based interfaces. Searching on Web browsers was the most common for all subjects, followed by library catalogues. Half had little or no experience of searching CD-ROM systems, commercial online systems or full-text databases.

More than three-quarters of searchers claimed to have very little or no knowledge of the topics. Just under a quarter said that topics 339 on 'Alzheimer's Drug Treatment ' and 347 on 'Wildlife Extinction' were marginally familiar.

2 Search Results.

2.1 Systems And Rounds.

The overall results for aspectual precision and recall for both systems are given in Figure 1. The breakdown by round in Figure 2 shows slightly higher precision in round one and slightly higher recall in round two.

Comparative results (Figure 3) also show higher precision for both rounds for searches on ZPrise and higher recall for those on Okapi. Searches on Okapi achieved better precision in round one and only marginally better recall in round two.

*Centre for Interactive Systems Research, Department of Information Science, City University, Northampton Square, London EC1V 0HB, UK

System	Measure	Mean	Median	Variance	Range
Okapi	Recall	0.400	0.391	0.082	0.000 – 1.000
	Precision	0.706	0.833	0.098	0.000 – 1.000
ZPrise	Recall	0.381	0.261	0.107	0.000 – 1.000
	Precision	0.809	0.877	0.081	0.000 – 1.000

Figure 1: Recall And Precision By System

Round	Measure	Mean	Median	Variance	Range
1	Recall	0.385	0.317	0.102	0.000 – 1.000
	Precision	0.784	0.868	0.080	0.000 – 1.000
2	Recall	0.396	0.359	0.087	0.000 – 1.000
	Precision	0.731	0.833	0.103	0.000 – 1.000

Figure 2: Recall And Precision By Round.

2.2 Individual Topics.

Figure 4 presents results for individual topics for each system ranked by aspectual precision and recall as given in Appendix D. The ranking by recall for both systems is almost identical except for topics 307 'New Hydroelectric Projects' and 347 'Wildlife Extinction', which are in position 4 and 5 in Okapi and reverse order in ZPrise. Topic 322 'International Art Crimes' ranks the lowest in both measures for both systems. In terms of precision, the greatest discrepancies is between topics 339 'Alzheimer's Drug Treatment' and 303 'Hubble Telescope Achievements', which rank 1 and 5 in Okapi and 4 and 2 in ZPrise respectively.

2.3 Searchers.

A comparison of searcher performance on the different systems (Appendix D.1) indicates that searchers in round one performed equally well on both system with half getting better precision/recall on either Okapi or ZPrise. However in round two all four searchers achieved higher precision on ZPrise, whereas recall was comparable between the two systems.

There is no evidence to show that the options introduced in the Okapi interface in round two had any effect on search performance. The 'clear feedback' option was used only once on topic 322, and the 'clear query' option six times, five on topic 322 and once on 339. The results for these topics in Appendix D show that in both cases precision and recall were lower than in round one.

Round	System	Measure	Mean	Median	Variance	Range
1	Okapi	Recall	0.393	0.391	0.092	0.111 – 1.000
		Precision	0.756	0.868	0.083	0.200 – 1.000
	ZPrise	Recall	0.378	0.261	0.122	0.000 – 1.000
		Precision	0.811	0.917	0.083	0.000 – 1.000
2	Okapi	Recall	0.407	0.388	0.080	0.000 – 1.000
		Precision	0.655	0.829	0.117	0.000 – 1.000
	ZPrise	Recall	0.384	0.263	0.101	0.000 – 1.000
		Precision	0.807	0.877	0.086	0.000 – 1.000

Figure 3: Recall And Precision By Round And System.

	OK		ZP	
Rank	Precn.	Recall	Precn.	Recall
1	339	303	326	303
2	326	339	303	339
3	307	326	347	326
4	347	307	339	347
5	303	347	307	307
6	322	322	322	322

Figure 4: Results Of Individual Topics In Ranked Order.

3 Search Perceptions And Performance.

3.1 Searching task.

At the end of the experiment six searchers felt that they had had a good or very good understanding of the task involved and two had reservations. Only half the subjects noted meaningful aspects on the search worksheets and three left them blank. With regard to the typicality of the searching task, only one searcher expressed that the task was similar to the searching tasks normally performed whilst the rest felt it only had some similarity.

3.2 Search Difficulty.

Overall half of the searches were classed as easy, a quarter as marginally difficult and a quarter as difficult. However more searches were deemed to be easy on ZPrise than on Okapi, 58% compared to 38%. In particular the searches on topic 339 'Alzheimer's Drug Treatment' and 326 'Ferry Sinkings' were considered to be unanimously easy by the four searchers on ZPrise but only by one searcher on Okapi. The results show that both topics were ranked second and third in both systems respectively in terms of recall but precision and recall were higher on Okapi. The six topics searched across both systems were quite evenly distributed with a third covering less than 25% of aspects, a third between 25% and 50% and a third over 50%.

3.3 Search Satisfaction.

In spite of searchers' perception that more topics were easy to search on ZPrise, 50% of searches carried out on the Okapi system were deemed to have produced a satisfactory outcome compared to 37% on ZPrise. But an equal proportion (34%) were found to be not satisfactory for both systems. The greatest difference appeared in topic 303 'Hubble Telescope Achievements' and 347 'Wildlife Extinction', where searchers on the Okapi system were more satisfied with the search outcome. In both cases the results do not bear out the searchers' perception. Precision and recall for the two topics were higher for ZPrise.

3.4 Confidence In Coverage Of Aspects.

Searchers of the Okapi system expressed that they were slightly more confident that they had identified all the possible aspects for the topics searched than those using ZPrise, 33% compared to 29%. However overall across both systems, responses indicate that 31% were confident, 29% were marginally confident and 40% were not confident. Searchers were the least confident about topic 322 'International Art Crime' (7 searchers out of 8) and the most confident about topic 303 'Hubble Telescope Achievements'. The lack of confidence for topic 322 was confirmed by the poor results for both systems on both measures. On the other hand 303 achieved the highest recall in both systems particularly in ZPrise where precision and recall were both higher.

In addition it is worth noting that, although ZPrise searches on topic 339 'Alzheimer's Drug Treatment' were unanimously considered to be easy, the level of confidence was unanimously marginal. By contrast three out of four Okapi searchers found the topic marginally difficult but they were confident that they had covered all the aspects. The topic ranked second in both systems on recall but the results for precision were better on Okapi.

3.5 Searching Time.

A much higher percentage of Okapi searchers considered that they had enough time to carry out an effective search than those using the ZPrise system, 45% as opposed to 25%. For 17% of searches users claimed to have marginally enough time compared to 33% for ZPrise. For both systems topics 322 'International Art Crime' and 347 'Wildlife Extinction' appear to have been the most problematic. As already mentioned topic 322 produced poor results overall, whereas 347 ranked 4th and 5th on recall and 3rd and 4th on precision in ZPrise and Okapi respectively.

4 System Perception.

4.1 Ease Of Use.

Overall searchers appear to have perceived little difference in the ease of use of the two systems. Both systems were deemed to be difficult to use in only 8% of the searches. Okapi was classed as easy to use for 63% of the searches as compared to 59% for ZPrise.

4.2 Learnability.

Okapi was deemed to be easy to learn for 75% of the searches undertaken and marginally easy for 25%. In the case of ZPrise, the system was considered to be easy to learn for 46% of searches, marginally easy for 50% and not easy for 4%. Topic 303 'Hubble Telescope Achievements' showed the greatest discrepancy, with all four Okapi searchers but only one ZPrise searcher claiming the system to be easy to learn. Although Okapi searchers expressed a high degree of satisfaction with the search outcome for this topic, the results for ZPrise as already noted were better.

4.3 Understanding.

For 71% of searches on Okapi, searchers'divulged that they found the system easy to understand compared to 54% on ZPrise. The extent in which the relevance feedback process is made more visible in the Okapi system, could account for this difference in user perception.

5 Searching Behaviour.

Data on searching behaviour and the search process is presented in Appendix C.

5.1 Query Terms And Iterations.

As would be expected there was little difference in the number of initial query terms used in either systems but the mean for Okapi in Trec5 was almost double that of Trec6, 7.06 as opposed to 3.16. Equally there is little difference between the final query terms for Okapi (4.84) and ZPrise (3.79). The modification in the incremental query expansion facility in Trec6 would account for the fewer terms in the final query compared to the 18.21 mean in Trec5.

User defined terms for all iterations were also comparable for both systems, 6 for Okapi and 6.92 for ZPrise. Because of the incremental query expansion, Okapi searchers introduced less than half the number of terms in the query after the first iteration than in ZPrise (2.33, 5.54). They also removed query terms. The current implementation of incremental query expansion led to substantially fewer terms being removed than in Trec5, 2.54 compared to 25.22.

Although ZPrise searchers did not appear to have generated more query terms, they did undertake slightly more iterations than in Okapi, 4.67 compared to 3.38.

5.2 Documents 'Viewed' And 'Seen'.

Searchers in Okapi demonstrated similar behaviour in scrolling through hitlists as in Trec5, covering a mean of 51.92 items compared to 60.58 previously. They equally examined the same number of full records, (14.15, 14.13) but fewer than in ZPrise (20.96).

TREC	Minimum number of relevant documents	RSV Must Not Be Less Than two-thirds the average RSV of
5	2	the terms in the last working query formulation
6	3	all terms that have been in the working query

Figure 5: TREC 5 and 6 Incremental Query Expansion Conditions

5.3 Relevance Judgements.

Although ZPrise searchers examined more full records on average, they made fewer positive relevance judgements 4.67 as opposed to 5.83 in Okapi. Relevance judgements in Okapi were also predominantly made on the full record (4.58) rather than on the best passage only (1.25).

6 Summary Of Results And Conclusions.

6.1 System Performance.

Okapi with its relevance feedback and query expansion facility clearly favoured recall. But since only one searcher overode the facility as a last resort in one search only, there is no evidence to show that this option could affect search performance. Searchers tended to rely on the system and intervened in a minimalist way. No diagnostic analysis has yet been undertaken to determine the effectiveness of the incremental query expansion.

The conditions specified in Appendix A.3.2 (2b) for members of the set of relevance feedback (RF) terms to be included in the set of candidate terms for the working query were modified from those used in TREC 5. The conditions, summarised in figure 5, were altered because the TREC 5 conditions often produced large sets of candidate terms (possibly several hundred) which sometimes resulted in:

1. apparently (from the user's viewpoint) erratic changes to the working query, and/or
2. the user removing large numbers of terms from the working query while searching for suitable RF terms.

The modified threshold conditions resulted in a reduced number of candidate terms, thereby making fewer and less marked changes to the working query. However, if we compare the number of user-entered and RF terms used in queries (figure 6) we see that, with the exception of three searches – (p24, t303i), (p11, t326i) and (p23, t326i) – very few RF terms were used by searchers. In 15 of the searches none were used at all. This may have been because user's saw them but removed them, or, more likely, few RF terms were determined by Okapi as suitable candidate terms. Clearly we need to carry out further experiments to:

1. establish more suitable conditions to provide greater recall and precision.
2. determine the usefulness of relevance feedback in a task of this nature.

6.2 Searcher Characteristics.

The sample of searchers appeared to be homogeneous and fairly representative of end-users. They all had a degree of experience with Web based systems, basic general knowledge and some awareness of advanced retrieval principles. The test systems were sufficiently different so that no individual subject had an unfair advantage for searching.

6.3 The Topics.

The sample of topics also seemed to have been evenly distributed between difficult, marginally difficult and easy searches. Topic 303 'Hubble Telescope Achievements' ranked first in terms of recall in both systems, and in terms of precision topic 326 'Ferry sinkings' in ZPrise and topic 339 'Alzheimer drug Treatment' in Okapi ranked first. There is no direct evidence to show that the number of aspects associated with any one topic determined the level

Topic	Searcher	User	RF	Topic	Searcher	User	RF
303	p12	2	0	326	p11	6	9
	p14	8	0		p13	3	4
	p22	6	0		p21	1	3
	p24	5	7		p23	3	15
307	p11	4	1	339	p12	3	0
	p13	2	0		p14	8	0
	p21	2	1		p22	4	0
	p23	2	0		p24	8	0
322	p11	10	1	347	p12	2	0
	p13	7	0		p14	5	3
	p21	10	0		p22	4	0
	p23	6	0		p24	6	0

Figure 6: Okapi: Number of User-Entered and RF Terms By Topic/Searcher

of difficulty. Whilst searchers could largely predict the performance of the easiest and most difficult topics, i.e. 322 'International Art Crime' they were more uncertain about those in between. On the whole there appeared to be no strong correspondence between searchers' perception about the topics and the system and the actual results.

6.4 Searchers' System Preferences.

Searchers found more searches to be easier on ZPrise but overall both systems were considered to be equally easy to use. Nevertheless searchers were more satisfied and confident about search outcomes with Okapi. The apparent discrepancies and inconsistencies between users' perception of the system and the searches would seem to indicate that searchers can differentiate between the procedural and more conceptual aspects of using a system or carrying out a search. However, there is clearly a tension between the two. It could be argued that the more interactive environment in Okapi offered the user the possibility for greater control in the searching process. Even though there was more to learn, the functionality was more apparent and made learnability easier. However, when asked which system they preferred to use, five out of the eight subjects opted for ZPrise.

References

[1] Robertson S E and Sparck Jones K. Relevance weighting of search terms. *Journal of the American Society for Information Science 27* May–June 1976 p129–146.

[2] Robertson S E. On term selection for query expansion. *Journal of Documentation 46* Dec 1990 p359–364.

[3] Robertson S E *et al.* Okapi at TREC–2. In: [6]. p21–34.

[4] Robertson S E *et al.* Okapi at TREC–3. In: [7]. p109–126.

[5] Robertson S E *et al.* Okapi at TREC–5. In: [9].

[6] *The Second Text REtrieval Conference (TREC–2).* Edited by D K Harman. Gaithersburg, MD: NIST, 1994.

[7] *Overview of the Third Text REtrieval Conference (TREC–3).* Edited by D K Harman. Gaithersburg, MD: NIST, April 1995.

[8] *Overview of the Fourth Text REtrieval Conference (TREC–4).* Edited by D K Harman. Gaithersburg, MD: NIST, November 1996.

[9] [TREC–5 proceedings.] NIST. To be published 1997.

A Okapi Interactive System Description

A.1 Introduction

A.1.1 The Okapi Interactive Interface.

The Okapi interface is an adaptation of that used in TREC 5 [5, Appendix A]. Figures 7 and 8 show screen dumps of the running system. The two major differences implemented for TREC 6 were:

1. A modification to the incremental query expansion conditions (see Section A.3.2).
2. Additional query manipulation facilities in the second round of searches (see Section A.4).

A.2 The Structure Of The GUI.

The interface is composed of: (1) a main window (figure 7) divided into six areas, and (2) a pop-up window in which full documents are displayed.

A.2.1 Main Window

1. **Query Entry Box**

   A text entry widget for user input of query terms. See Section A.3.1.

2. **Working Query**

   A scrollable, ranked list of the terms in the current working query. See Section A.3.2.

3. **Removed Terms**

   A scrollable list of any terms removed by the user from the working query, displayed in removal order. See Section A.3.7

4. **Hitlist**

   A scrollable, ranked hitlist for the current iteration. See Section A.3.4.

5. **Pool of Positive Relevance Judgments**

   A scrollable, ranked list of positive user relevance judgments. See Section A.3.6

6. **Function Buttons**

   At the bottom of the window are either two or three context-sensitive buttons.

   Search See Section A.3.3.

   Exit See Section A.3.9.

   Query Options (Round 2 only). See Section A.4.

A.2.2 Full Document

A pop-up text window for the display of a full record selected from the hitlist. At the bottom of the window are three buttons for making relevance judgments. These are described in Section A.3.5.

A.3 User Interaction

A.3.1 User Input Of Query Terms

Users may enter one or more words into the query entry box followed by an optional "phrase" operator (a '+' sign) as the last non-space character in the line. All the terms entered will be stemmed and looked-up in the database; stopwords and non-indexed terms will be discarded. If the operator is:

1. **None**: Each non-stopped index term will be a single term in the query.
2. **Phrase (+)**: If all of the non-stopwords are index terms an Okapi "phrase" (which constitutes a single term) will be formed as follows.
   (a) Two sets will be formed:
   A: A phrase in term input order, possibly with intervening stopwords: n(A) postings, weight w(A).
   S: A "within same sentence" occurrence of the terms in any order: n(S) postings, weight w(S).
   (b) These are combined into one set — a single query term — using the bestmatch operator BM25 ([4, Section 3.2]) where appropriate, according to the following rules.

Sets Generated	Sets Used	Displayed Operator
1. n(A) = n(S) = 0	Both discarded	
2. n(A) = 0, n(S) > 0	S(S), n(S), w(S)	(S)
3. n(A) > 0, n(A) = n(S)	S(A), n(A), w(A)	None
4. 0 < n(A) < n(S)	S(A), n(A), w(A) and S(S)-S(A), n(S)-n(A), w(S)	(B)

The weight calculated for each term is a Robertson/Sparck-Jones F4 predictive weight, with halves [1]. The weighting function allows each user entered term to be assigned a "loaded" weight by assuming the existence of a set of "mythical rels" of which a fixed number contain the term. The number of "mythical rels" is called "bigrload" of which "rload" contain the term. The values used for "rload" and "bigrload" were 4 and 5 respectively.

A.3.2 The Working Query

The entire set of user-entered and system-generated terms (the "termset") will be referred to by the letter Q. The number of members of Q is n(Q). Using similar terminology there will, at any time during a search, exist the following subsets of Q.

U: n(U) user-entered terms.

E: n(E) non-user terms extracted during relevance feedback.

C: n(C) candidate terms, those members of Q that satisfy the query threshold conditions.

W: n(W) members of the working query.

After each change to Q the working query will be re-generated in the following three stages.

1. Q is sorted by the two keys:
   (a) USER_TYPE: U (user) or E (extracted), descending.
   (b) RSV: (Robertson Selection Value [2]) descending
2. C is formed from Q by taking:
   (a) All members of U.
   (b) Members of E that occur in at least three relevant documents and have an RSV >= two-thirds the average RSV of all terms that have been in W.
3. W is formed from the top N members of C (N <= MAX_TERMS, a system defined limit), i.e. all user-entered terms plus and the top {MAX_TERMS - n(U)} members of E. If n(U) >= MAX_TERMS then W will be made up of members of U only. In both rounds MAX_TERMS defaulted to 20. In the second round users were able to increase its value to 30 or 40 if they wished, although no users did so.

NOTE: The conditions specified in A.3.2 (2b) were modified from those used in TREC 5. In TREC 5 the non-U members of C had to occur in at least two relevant documents and have an RSV >= two-thirds the average RSV of the terms in the *LAST WORKING QUERY FORMULATION*. These conditions (sometimes) produced large sets of candidate terms (possibly several hundred) which either caused some erratic behaviour (from the user's viewpoint) in changes to the working query, and/or resulted in the user removing large numbers of terms from W while searching for suitable query terms. The modified threshold conditions resulted in a reduced value of n(C), thereby making fewer and less marked changes to the working query.

Terms are displayed in the working query window in descending order of RSV.

A.3.3 Searching The Database.

Each search, performed by clicking the "Search" button, marks the next iteration. The members of W are combined using a best match operation (bm25). Passage retrieval [3, 4] is applied to the document set generated with parameters p_unit = 4, p_step = 2, k1 = 1.6 and b = 0.7. This will result in the system finding two passages for each document.

1. The full document: weight = w(F), length = l(F).
2. A sub-passage of the document: weight = w(P), length = l(P).

There are two cases to consider.

1. l(F) = l(P) :: The passage and the full document are the same and w(F) = w(P).
2. l(F) > l(P) :: The passage is distinct from the full document; the weight assigned to the document will be the greater of w(F) and w(P).

A.3.4 Hitlist Generation

Searching the database will result in a hitlist, generated from the ranked document set, displayed in the hitlist window — A.2.1(4). The hitlist is made up of the top H ranked documents (H <= 50). Each document must satisfy the following conditions.

1. It is not a member of the current set of relevant documents (i.e. it must not have already been seen in full by the searcher.
2. l(P) (or l(F) if there is no passage) must be less than DOC_THRESHOLD characters in length. DOC_THRESHOLD defaults to 10K.

An entry for each document consists of:

- A header line.

  ⟨record_no⟩ ⟨docid⟩ ⟨normalised_weight⟩[⟨passage_length⟩/] ⟨document_length⟩

  The ⟨normalised_weight⟩ is the system weight mapped onto the range 1..1000. ⟨document_length⟩ and ⟨passage_length⟩ are given to the nearest page, where a page is taken to be 2000 characters.

- A system generated title, made up from approximately the first 150 characters from the start of the document.
- Query term occurrence information

  A count of the occurrences of the stems of each query term within the document. The stem of a query term may occur in different source forms within the document. The first source in the document for each stem will be shown.

The hitlist entry at the top of the window is the first unseen document in the list.

A.3.5 Showing Documents

Double-clicking anywhere in a document's hitlist entry will display the full document in a pop-up, scrollable text window. Query terms are highlighted in green. The passage (if there is one) is highlighted in light grey. The line of the document displayed to the user at the top of the window is dependent upon the values of w(F|P) and l(F|P) (see Section A.3.3) i.e.

1. l(F) = l(P): Don't highlight the passage. The line of the document at the top of the window is the first line containing a query term.
2. l(F) > l(P): Highlight the passage. The line of the document at the top of the window is:
   (a) If w(P) >= w(F):: the first line of the passage.
   (b) If w(P) < w(F):: the first line containing a query term.

At the bottom of the text window are three buttons to allow users to make a relevance judgment.

1. **Full Document**: The document contains one or more aspects. Extract terms from the entire document.
2. **Passage Only**: The document contains one or more aspects. Extract terms from the highlighted passage only.
3. **Not Relevant**: The document does not contain any aspects.

Searchers must make a relevance judgment before they may go onto to any other part of the search process. For a document shown from the current hitlist a relevance judgment can be altered at any time until a new search is made. The change, (F -> P), (P -> F), ([F | P] -> N), or (N -> [F | P]) will modify the set of extracted terms appropriately.

A.3.6 Relevance Judgments Pool

The ranked hitlist information for all documents currently judged as relevant. Any member of the current relevance judgments pool that exists in a document set for a new iteration has its weight set to its value in the latest document set; the display order is adjusted accordingly.

A.3.7 Removing Terms

Any term may be removed from the working query by double clicking on its entry in the query window. Removed terms are displayed in the removed terms window (A.2.1(3)). If n(C) > MAX_TERMS (Section A.3.2), as terms are removed, other terms may be promoted to take their place.

A.3.8 Reinstating Removed Terms

A removed term may be reinstated in the query by double-clicking on its entry in the removed_terms window, although, as the working query changes, (i) its rank position may be different from that when it was removed, and (ii) in the case of extracted terms, it may not go in at all if n(C) >= MAX_TERMS.

A.3.9 Quitting

Quitting the search is achieved by clicking once on the "Exit" button.

A.4 Changes To The Interface For The Second Round Of Searches.

The functionality of the interface used in the first round of searches, with the modifications to the incremental expansion of the working query as described in Section A.3.2, was identical to that used in TREC 5. For the second round additional functionality was added as a direct result of observing users in Round 1 (a) attempting to change track during some searches (particularly 322i) and (b) obtaining during the course of a search a set of candidate terms (C) greater than 20 — the default value of MAX_TERMS — in size.

The additional functionality was implemented by installing a third function button (see Section A.2.1(6)) on the Main Window. Clicking on this button initiates a pop-up menu containing the entries:

1. Clear Relevance Feedback.
2. Clear Working Query.
3. Set Working Query Size

A.4.1 Clear Relevance Feedback

1. The set of terms added automatically to Q by the system after each positive relevance judgement are "removed" from Q (the entire set of user-entered and system-generated terms). The terms in fact remain in Q but their values of little r (the number of relevant documents they occur in) are set to zero and they are "flagged" to indicate that they should not be members of the set of candidate terms (C), unless subsequently:
   (a) re-input by the user, or
   (b) satisfy the threshold conditions based upon new relevance judgements (Section A.3.2)
2. The current set of relevance judgements would appear "greyed-out" in the relspool window (A.2.1(5)) to indicate that they were made before the relevance feedback was cleared. As before, these documents are not allowed to be members of any subsequent hitlist.

A.4.2 Clear Working Query

In addition to the "removal" of relevance feedback terms (Section A.4.1), all user-entered terms are similarly "removed".

A.4.3 Set Working Query Size

The default maximum size of the working query (MAX_TERMS) is 20 terms. In general, as a result of the modified threshold conditions for membership of the set of candidate terms (see Section A.3.2), it was found that n(C) did not often exceed this value. However, if n(C) is > 20 the only way the user can see the extra terms in the set is by removing one or more of the top 20 terms so that lower terms are promoted.

This option allows the user to increase the maximum working query size to 30 or 40 terms thus:

1. Enabling the user to see more than the top 20 terms from C without having to remove terms from W,
2. Allowing the user to increase the number of terms used to search the database if relevance feedback is producing enough good terms to warrant this.

During the second round of searches the additional functionality was seldom used (see Section C.7). The "clear working query" and "clear relevance feedback" facilities were only used a limited number of times, mostly on topic 322i which all users found difficult. The "Set Working Query" option was not used.

B Experimental Conditions

B.1 Searcher characteristics

Eight searchers, all MSc students in the Department Of Information Science, were divided into two groups: Group 1 and Group 2. Group 1 consisted of one male and three females; Group 2 consisted of three males and one female. Their ages ranged from late 20s to early 40s. The searchers were end-users who had:

1. No specialist knowledge of any of the subject domains covered by the topics, and
2. Only theoretical knowledge of relevance feedback IR systems.

B.2 Task Description / Training.

The task for each searcher consisted of:

1. Training.

   Each searcher conducted a tutorial session for each IR system separately from their search session. This was conducted in the week preceding the search session.

2. Search Session.

   (a) Filling in a pre-session questionnaire.

   (b) Reading a set of introductory instructions which outlined the nature of the task.

   (c) Performing six searches on the assigned topics (in the order 326i, 322i, 307i, 347i, 303i and 339i), three on Okapi (OK) and three on the control IR system, ZPrise (ZP). The searcher-system-topic combinations were:

Searcher	326i	322i	307i	347i	303i	339i
1	OK	OK	OK	ZP	ZP	ZP
2	ZP	ZP	ZP	OK	OK	OK
3	OK	OK	OK	ZP	ZP	ZP
4	ZP	ZP	ZP	OK	OK	OK

   After each search a post-search questionnaire was completed.

   (d) Filling in a post-session questionnaire.

   Each search was to take a maximum of 20 minutes which include reading time for the topic as well as actual searching time.

C Search Process

Unless otherwise stated the column 'Type' entries **N**, **+** and **A** refer to:

N terms defined with no adjacency operator
+ terms defined with an adjacency operator
A all terms defined

C.1 Clock Time

Times are given to the nearest tenth of a minute.

System	Mean	Median	Variance	Range
Okapi	19.8	20.5	3.20	15.3–21.8
ZPrise	20.3	20.7	2.61	16.1–22.5

C.2 Number of User Defined Terms

C.2.1 At All Iterations

System	Type	Mean	Median	Variance	Range
Okapi	N	4.04	3.0	13.17	0–16
	+	1.96	1.0	6.82	0– 9
	A	6.00	4.5	24.61	1–25
ZPrise		6.92	6.0	15.99	2-17

C.2.2 After The First Iteration

System	Type	Mean	Median	Variance	Range
Okapi	N	1.08	0.0	5.99	0–10
	+	1.25	0.0	5.15	0– 8
	A	2.33	0.5 5	14.84	0–17
ZPrise		5.54	5.5	20.26	0-15

C.2.3 "Phrases" Defined By Searchers (Okapi only)

Phrases generated: 47
Phrases used: 38

C.3 Number of Terms Used In Queries

C.3.1 Initial Query

System	Type	Mean	Median	Variance	Range
Okapi	N	2.58	3.0	3.21	0–8
	+	0.58	0.0	0.51	0–2
	A	3.16	3.0	2.67	2–8
ZPrise		3.58	3.0	3.21	2- 8

C.3.2 Final Query

System	Type	Mean	Median	Variance	Range
Okapi	N	3.63	3.0	13.03	0–13
	+	1.21	1.0	2.52	0– 7
	A	4.84	4.0	12.93	1–15
ZPrise		3.79	3.5	4.61	1-10

C.4 Number of Iterations

An iteration, i.e. a new query formulation, was taken to be marked by each 'search' command.

System	Mean	Median	Variance	Range
Okapi	3.38	2.0	7.11	1- 9
ZPrise	4.67	4.0	14.93	1-17

Okapi note: Expansion was performed incrementally. No data was kept to indicate how many times and when each working query was altered by the inclusion/exclusion of extracted terms.

C.5 Documents "Viewed" (hitlist) and "Seen" (full text)

C.5.1 Viewed

"Viewing" a document consisted of seeing its entry in the hitlist but not the full document. The figures, for Okapi only, represent the percentage distance scrolled through the hitlist by the searcher. There was no information available from the ZPrise log files.

System	Mean	Median	Variance	Range
Okapi	51.92	50.0	611.30	14–98

C.5.2 Seen.

"Seeing" a document consisted of showing the full record. In the hitlist for any iteration Okapi does not show documents that were included in the hitlist(s) of any previous iteration(s) (see Section A.3.4. This is not the case with ZPrise where the same document may have occurred in several hitlists. Two sets of figures are shown: (a) all shows, including repeated shows, and (b) distinct shows. All includes some documents that were shown more than once with a view to (potentially) changing the relevance judgement.

System		Mean	Median	Variance	Range
Okapi	Distinct	13.58	13.5	31.38	5-30
	All	14.25	14.5	29.33	7-30
ZPrise	Distinct	15.92	15.0	83.73	2-32
	All	20.96	20.0	165.43	3-52

C.6 Relevance Judgments

Documents may have been judged more than once during a session, either altering the judgement or leaving it unchanged. In an Okapi session this may only happen during a single iteration due to the conditions specified in Section A.3.4. In a ZPrise session, since individual documents may occur in several hitlists this may occur at any time during the session.

Judging a document relevant implies that it contains one or more aspects. The types of relevance judgement for Okapi (F, P and N) are described in A.3.5. ZPrise has two judgements: R, relevant, means that the document contains one or more aspects. U, undecided, means that the document does not contain any aspects.

Two sets of figures are given, one for the final judgement and one for the total judgements on each document.

1. Final judgement only.

System	Rel	Mean	Median	Variance	Range
Okapi	F	4.58	4.0	11.56	1-11
	P	1.25	1.0	2.37	0- 6
	N	7.96	6.0	23.78	3-24
	All	13.79	13.5	29.91	6-30
ZPrise	R	4.67	2.0	34.41	0-26
	U	0.08	0.0	0.08	0- 1
	All	4.75	2.0	33.67	1-26

2. All Judgements.

The figures for "All Judgements" include the following re-judgements of documents.

System	(F->N) or (R->U)	(N->F) or (U->R)	Unchanged	Total
Okapi	2	4	9	15
ZPrise	1	1	2	4

System	Rel	Mean	Median	Variance	Range
Okapi	F	4.67	4.0	11.80	1-11
	P	1.25	1.0	2.37	0- 6
	N	8.33	6.5	23.10	3-24
	All	14.25	14.5	29.33	7-30
ZPrise	R	4.79	2.5	34.95	0-26
	U	0.13	0.0	0.11	0- 1
	All	4.92	3.0	34.43	0-26

C.7 Use of System Features

The figures in parentheses for Okapi are the equivalent values for Trec 5.

System	Command		Mean	Median	Variance	Range
Okapi	Define	N	4.04 (4.96)	3.0 (3.5)	13.17 (21.58)	0-16 (0-16)
		+	1.96 (3.83)	1.0 (3.0)	6.82 (9.80)	0-9 (0-13)
		A	6.00 (8.79)	4.5 (8.0)	24.61 (24.96)	1-25 (2-18)
	Search		3.38 (3.63)	2.0 (3.0)	7.11 (1.98)	1- 9 (1- 7)
	Show		14.15 (14.13)	14.5 (13.5)	29.33 (15.42)	7-30 (9- 21)
	Remove		2.54 (25.22)	0.5 (13.5)	12.00 (1119.18)	0-11 (0-116)
	Restore		0.38 (0.85)	0.0 (0.0)	1.64 (3.40)	0- 6 (0- 8)
	Clear_rf		0.08	0.0	0.08	* 0- 1
	Clear_query		0.58	0.0	2.08	** 0- 5

* Topic 322i

* Topic 322i (6 times, 5 by one searcher)
Topic 339i once.

System	Command	Mean	Median	Variance	Range
ZPrise	Define	6.92	6.0	15.99	2-17
	Search	4.67	4.0	14.93	1-17
	Show	15.92	15.0	83.73	2-32

C.8 Number of User Errors

Data on user errors were not collected.

C.9 Topic 326 - Ferry Sinkings.

C.9.1 Search Narrative

The initial query terms entered were chosen from the topic specification itself. Eight documents were viewed from the initial hitlist but only four were deemed to meet the criteria set out by the narrative. One was chosen after subsequent re-examination. As some of the documents selected from the first hitlist referred to the sinking of the ferry 'Herald of Free Enterprise' without giving the number of deaths involved the term 'Herald of Free Enterprise' was added for the second iteration. Four of the five documents viewed from the second hitlist were judged to be relevant. The term 'Herald of Free Enterprise' was then removed and the term 'Manila ferry' added for a third iteration. This was in order to find other documents that dealt more specifically with the sinking of this ferry, but this failed to retrieve any new documents that were relevant.

As the query terms used in each search is reflected in the weighted list of documents retrieved the continued inclusion of the search term '100 deaths' may not have been particularly useful for the second and third iteration

and on reflection should, perhaps, have been removed from the term set. In retrospect the use of the search phrase '100 deaths' may have contributed to some relevant documents having a lower ranking on the hitlist for all three iterations. The display of extracted query terms did not prove to be particularly useful and the inclusion of the search terms 'ferry' and 'sinks' as well as the search phrase 'ferry sinking' may not have been very helpful either. In all cases, but one, in making positive relevance judgements, the searcher chose to select the full document rather than just the highlighted passage. Due to the time constraints the searcher did not review the final list of documents selected as relevant in order to weed out those which may have duplicated reports of any particular ferry sinking.

The searcher was not very happy with the way in which relevance judgements had to be made that were, upon subsequent iterations, irretrievable. The searcher would have also preferred that on iterations involving a newly defined search term or phrase that there would have to be positive selection of existing query terms to be included in the new search instead of the automatic inclusion that did occur. The need for selection or de-selection of query terms required for each new iteration was found to be cumbersome.

The search itself was not particularly difficult. The searcher did experience some difficulty in introducing a search term that would help select reports of a ferry sinking that involved the loss of more than 100 lives and this may have led to some problems later in the search. It may have been better to simplify the initial search by restricting the query to 'ferry sinking' or 'ferry' and 'sinks'.

C.9.2 Breakdown of command usage.

Define (N)	0
Define (+)	4
Search	3
Show	16
Remove	1

D City Recall-Precision Results

D.1 Results Grouped By Round, Searcher And System.

D.1.1 Recall

Round	Searcher	System	Mean	Median	Variance	Range
1	p11	Okapi	0.315	0.391	0.032	0.111 – 0.444
	p12	Okapi	0.672	0.900	0.235	0.115 – 1.000
	p13	Okapi	0.353	0.391	0.051	0.111 – 0.556
	p14	Okapi	0.232	0.154	0.021	0.143 – 0.400
	p11	ZPrise	0.631	0.700	0.167	0.192 – 1.000
	p12	ZPrise	0.140	0.111	0.005	0.087 – 0.222
	p13	ZPrise	0.600	0.500	0.130	0.300 – 1.000
	p14	ZPrise	0.140	0.087	0.030	0.000 – 0.333
2	p21	Okapi	0.206	0.174	0.050	0.000 – 0.444
	p22	Okapi	0.544	0.400	0.163	0.231 – 1.000
	p23	Okapi	0.278	0.333	0.022	0.111 – 0.391
	p24	Okapi	0.600	0.700	0.035	0.385 – 0.714
	p21	ZPrise	0.464	0.600	0.115	0.077 – 0.714
	p22	ZPrise	0.169	0.174	0.003	0.111 – 0.222
	p23	ZPrise	0.618	0.700	0.184	0.154 – 1.000
	p24	ZPrise	0.287	0.304	0.078	0.000 – 0.556

D.1.2 Precision.

Round	Searcher	System	Mean	Median	Variance	Range
1	p11	Okapi	0.807	0.846	0.016	0.667 – 0.909
	p12	Okapi	1.000	1.000	0.000	1.000 – 1.000
	p13	Okapi	0.641	0.833	0.146	0.200 – 0.889
	p14	Okapi	0.578	0.400	0.135	0.333 – 1.000
	p11	ZPrise	0.861	0.833	0.016	0.750 – 1.000
	p12	ZPrise	1.000	1.000	0.000	1.000 – 1.000
	p13	ZPrise	0.826	0.812	0.028	0.667 – 1.000
	p14	ZPrise	0.556	0.667	0.259	0.000 – 1.000
2	p21	Okapi	0.611	0.833	0.287	0.000 – 1.000
	p22	Okapi	0.861	0.833	0.016	0.750 – 1.000
	p23	Okapi	0.675	0.824	0.177	0.200 – 1.000
	p24	Okapi	0.472	0.500	0.044	0.250 – 0.667
	p21	ZPrise	1.000	1.000	0.000	1.000 – 1.000
	p22	ZPrise	0.933	1.000	0.013	0.800 – 1.000
	p23	ZPrise	0.722	0.833	0.037	0.500 – 0.833
	p24	ZPrise	0.573	0.800	0.250	0.000 – 0.920

D.2 Results Grouped by "System", "Round" and "Topic"

D.2.1 Recall.

System	Round	Topic	Mean	Median	Variance	Range
Okapi	1	303	0.572	0.572	0.367	0.143 – 1.000
		307	0.391	0.391	0.000	0.391 – 0.391
		322	0.111	0.111	0.000	0.111 – 0.111
		326	0.500	0.500	0.006	0.444 – 0.556
		339	0.650	0.650	0.125	0.400 – 0.900
		347	0.135	0.135	0.001	0.115 – 0.154
	2	303	0.857	0.857	0.041	0.714 – 1.000
		307	0.283	0.283	0.024	0.174 – 0.391
		322	0.056	0.056	0.006	0.000 – 0.111
		326	0.389	0.389	0.006	0.333 – 0.444
		339	0.550	0.550	0.045	0.400 – 0.700
		347	0.308	0.308	0.012	0.231 – 0.385
ZPrise	1	303	1.000	1.000	0.000	1.000 – 1.000
		307	0.087	0.087	0.000	0.087 – 0.087
		322	0.056	0.056	0.006	0.000 – 0.111
		326	0.278	0.278	0.006	0.222 – 0.333
		339	0.500	0.500	0.080	0.300 – 0.700
		347	0.346	0.346	0.047	0.192 – 0.500
	2	303	0.857	0.857	0.041	0.714 – 1.000
		307	0.239	0.239	0.008	0.174 – 0.304
		322	0.056	0.056	0.006	0.000 – 0.111
		326	0.389	0.389	0.056	0.222 – 0.556
		339	0.650	0.650	0.005	0.600 – 0.700
		347	0.116	0.116	0.003	0.077 – 0.154

D.2.2 Precision.

System	Round	Topic	Mean	Median	Variance	Range
Okapi	1	303	0.667	0.667	0.222	0.333 – 1.000
		307	0.840	0.840	0.000	0.833 – 0.846
		322	0.434	0.434	0.109	0.200 – 0.667
		326	0.899	0.899	0.000	0.889 – 0.909
		339	1.000	1.000	0.000	1.000 – 1.000
		347	0.700	0.700	0.180	0.400 – 1.000
	2	303	0.500	0.500	0.125	0.250 – 0.750
		307	0.912	0.912	0.015	0.824 – 1.000
		322	0.100	0.100	0.020	0.000 – 0.200
		326	0.917	0.917	0.014	0.833 – 1.000
		339	0.834	0.834	0.055	0.667 – 1.000
		347	0.667	0.667	0.055	0.500 – 0.833

System	Round	Topic	Mean	Median	Variance	Range
ZPrise	1	303	1.000	1.000	0.000	1.000 – 1.000
		307	0.834	0.834	0.055	0.667 – 1.000
		322	0.500	0.500	0.500	0.000 – 1.000
		326	1.000	1.000	0.000	1.000 – 1.000
		339	0.750	0.750	0.014	0.667 – 0.833
		347	0.781	0.781	0.002	0.750 – 0.812
	2	303	0.750	0.750	0.125	0.500 – 1.000
		307	0.800	0.800	0.000	0.800 – 0.800
		322	0.500	0.500	0.500	0.000 – 1.000
		326	0.960	0.960	0.003	0.920 – 1.000
		339	0.917	0.917	0.014	0.833 – 1.000
		347	0.917	0.917	0.014	0.833 – 1.000

E Search Evaluation: Questionnaire Results

E.1 Pre-Session Questionnaire.

1. What is the highest degree you have or expect to obtain?

Degree:	Bachelors	Masters	PhD
	0	8	0

2. What is your age?

Range:	<21	21-30	31-40	41-50	51-60	¿60
	0	5	3	0	0	0

3. What is your gender (M = male, F = female)?

Gender:	M	F
	4	4

4. Have you participated in previous TREC searching studies (Y/N)?

Answer:	Y	N
	0	8

5. Have you experience of using Okapi or ZPrise?

System:	Okapi	ZPrise
	1	0

6. How many years have you been doing online searching?

Years:	0	1	2	3	4	5	6	7
	1	3	1	2	1	0	0	0

7. How much experience have you had of the following (1=none, 3=some, 5=a great deal)

	1	2	3	4	5
Searching on computerised library catalogues	1	1	3	3	0
Searching on CD ROM systems	3	1	3	1	0
(e.g. Infotrac, Grolier) Searching on commercial online systems (e.g. Dialog, Lexis, BRS Afterdark)	3	1	2	1	1
Searching on WWW browsers (e.g. Netscape)	0	0	0	6	2
Searching on other systems	7	1	0	0	0
Searching full-text databases	3	2	1	2	0
Searching in ranked output IR systems	2	3	2	1	0
Searching in IR systems that provide relevance feedback	3	3	2	0	0
Using a mouse-based interface	0	0	2	2	4

E.2 Exit Questionnaire.

Rate the following on a scale of 1-5 (1=Not at all, 5=Completely)

1. To what extent were you able to understand the nature of the task?

Rating:	1	2	3	4	5
	0	0	2	5	1

2. To what extent did you find this task similar to other searching tasks that you typically perform?

Rating:	1	2	3	4	5
	0	2	5	1	0

3. How different did you find the following systems from each other?

Rating:	1	2	3	4	5
	0	0	4	4	0

4. Please rank the systems (1 = easiest).

	Okapi	ZPrise
Easiest to use:	4	4
Easiest to learn to use:	4	4
Liked best:	3	5

E.3 Post-Search Questionnaires.

The following questions were asked after each search session (i.e. after performing a search on one topic on one system).

1. How familiar are you with the topic?
2. How difficult was it to do the search?
3. How satisfied are you with your search results?
4. How confident are you that you identified all of the possible aspects for this topic?
5. Did you have enough time to do an effective search?
6. How easy was it to use this IR system?
7. How easy was it to learn to use this IR system?
8. How well did you understand how to use this IR system?

E.3.1 All Answers. (Percentages out of 48)

	1	2	3	4	5
1	60	19	19	2	0
2	25	23	29	19	4
3	19	15	23	38	5
4	25	15	29	27	4
5	17	23	25	23	12
6	0	8	31	46	15
7	0	2	38	42	18
8	0	0	38	60	2

E.3.2 Answers By System (Percentages out of 24).

System	Qu	1	2	3	4	5	System	Qu	1	2	3	4	5
Okapi	1	70	8	22	0	0	ZPrise	1	50	29	17	4	0
	2	25	13	36	22	4		2	25	33	21	17	4
	3	21	13	16	42	8		3	17	17	29	33	4
	4	25	13	29	29	4		4	25	17	29	25	4
	5	17	21	17	21	24		5	17	25	33	25	0
	6	0	8	29	46	17		6	0	8	33	46	13
	7	0	0	25	58	17		7	0	4	50	25	21
	8	0	0	29	67	4		8	0	0	46	54	0

E.3.3 Answers By Topic (Percentages out of 8).

Topic	Qu	1	2	3	4	5	Topic	Qu	1	2	3	4	5
303	1	50.0	37.5	12.5	0.0	0.0	326	1	62.5	25.0	12.5	0.0	0.0
	2	12.5	12.5	62.5	12.5	0.0		2	37.5	25.0	37.5	0.0	0.0
	3	12.5	12.5	37.5	37.5	0.0		3	12.5	0.0	37.5	37.5	12.5
	4	12.5	25.0	12.5	50.0	0.0		4	12.5	12.5	50.0	25.0	0.0
	5	12.5	0.0	37.5	37.5	12.5		5	12.5	12.5	50.0	12.5	12.5
	6	0.0	0.0	62.5	25.0	12.5		6	0.0	0.0	25.0	50.0	25.0
	7	0.0	0.0	37.5	50.0	12.5		7	0.0	0.0	25.0	50.0	25.0
	8	0.0	0.0	50.0	50.0	0.0		8	0.0	0.0	25.0	62.5	12.5
307	1	87.5	0.0	12.5	0.0	0.0	339	1	37.5	12.5	37.5	12.5	0.0
	2	37.5	25.0	12.5	25.0	0.0		2	25.0	37.5	37.5	0.0	0.0
	3	0.0	12.5	37.5	50.0	0.0		3	0.0	12.5	12.5	62.5	12.5
	4	12.5	25.0	37.5	12.5	12.5		4	0.0	0.0	62.5	37.5	0.0
	5	12.5	37.5	12.5	25.0	12.5		5	0.0	25.0	37.5	25.0	12.5
	6	0.0	0.0	25.0	62.5	12.5		6	0.0	0.0	37.5	50.0	12.5
	7	0.0	0.0	37.5	37.5	25.0		7	0.0	0.0	37.5	50.0	12.5
	8	0.0	0.0	25.0	75.0	0.0		8	0.0	0.0	37.5	62.5	0.0
322	1	87.5	12.5	0.0	0.0	0.0	347	1	37.5	25.0	37.5	0.0	0.0
	2	12.5	0.0	12.5	50.0	25.0		2	25.0	37.5	12.5	25.0	0.0
	3	62.5	25.0	12.5	0.0	0.0		3	25.0	25.0	0.0	37.5	12.5
	4	75.0	12.5	0.0	12.5	0.0		4	37.5	12.5	12.5	25.0	12.5
	5	25.0	37.5	12.5	12.5	12.5		5	37.5	25.0	0.0	25.0	12.5
	6	0.0	25.0	12.5	50.0	12.5		6	0.0	25.0	25.0	37.5	12.5
	7	0.0	12.5	37.5	25.0	25.0		7	0.0	0.0	50.0	37.5	12.5
	8	0.0	0.0	37.5	62.5	0.0		8	0.0	0.0	50.0	50.0	0.0

E.3.4 Answers By System–Topic (Percentages out of 4).

System	Topic	Qu	1	2	3	4	5	Topic	Qu	1	2	3	4	5
Okapi	303	1	75	25	0	0	0	326	1	75	0	25	0	0
		2	25	0	50	25	0		2	0	25	75	0	0
		3	0	0	50	50	0		3	25	0	50	25	0
		4	0	25	0	75	0		4	25	25	50	0	0
		5	0	0	25	50	25		5	25	25	25	0	25
		6	0	0	50	25	25		6	0	0	50	25	25
		7	0	0	0	75	25		7	0	0	25	50	25
		8	0	0	50	50	0		8	0	0	25	50	25
Okapi	307	1	100	0	0	0	0	339	1	50	0	50	0	0
		2	50	25	25	0	0		2	25	0	75	0	0
		3	0	25	0	75	0		3	0	25	0	50	25
		4	0	25	75	0	0		4	0	0	25	75	0
		5	0	50	0	25	25		5	0	0	50	25	25
		6	0	0	25	75	0		6	0	0	25	50	25
		7	0	0	25	75	0		7	0	0	25	50	25
		8	0	0	0	100	0		8	0	0	25	75	0

System	Topic	Qu	1	2	3	4	5	Topic	Qu	1	2	3	4	5
Okapi	322	1	100	0	0	0	0	347	1	25	25	50	0	0
		2	0	0	0	75	25		2	50	25	0	25	0
		3	75	25	0	0	0		3	25	0	0	50	25
		4	100	0	0	0	0		4	25	0	25	25	25
		5	50	25	0	0	25		5	25	25	0	25	25
		6	0	25	25	50	0		6	0	25	0	50	25
		7	0	0	50	50	0		7	0	0	25	50	25
		8	0	0	25	75	0		8	0	0	50	50	0
ZPrise	303	1	25	50	25	0	0	326	1	50	50	0	0	0
		2	0	25	75	0	0		2	75	25	0	0	0
		3	25	25	25	25	0		3	0	0	25	50	25
		4	25	25	25	25	0		4	0	0	50	50	0
		5	25	0	50	25	0		5	0	0	75	25	0
		6	0	0	75	25	0		6	0	0	0	75	25
		7	0	0	75	25	0		7	0	0	25	50	25
		8	0	0	50	50	0		8	0	0	25	75	0
ZPrise	307	1	75	0	25	0	0	339	1	25	25	25	25	0
		2	25	25	0	50	0		2	25	75	0	0	0
		3	0	0	75	25	0		3	0	0	25	75	0
		4	25	25	0	25	25		4	0	0	100	0	0
		5	25	25	25	25	0		5	0	50	25	25	0
		6	0	0	25	50	25		6	0	0	50	50	0
		7	0	0	50	0	50		7	0	0	50	50	0
		8	0	0	50	50	0		8	0	0	50	50	0
ZPrise	322	1	75	25	0	0	0	347	1	50	25	25	0	0
		2	25	0	25	25	25		2	0	50	25	25	0
		3	50	25	25	0	0		3	25	50	0	25	0
		4	50	25	0	25	0		4	50	25	0	25	0
		5	0	50	25	25	0		5	50	25	0	25	0
		6	0	25	0	50	25		6	0	25	50	25	0
		7	0	25	25	0	50		7	0	0	75	25	0
		8	0	0	50	50	0		8	0	0	50	50	0

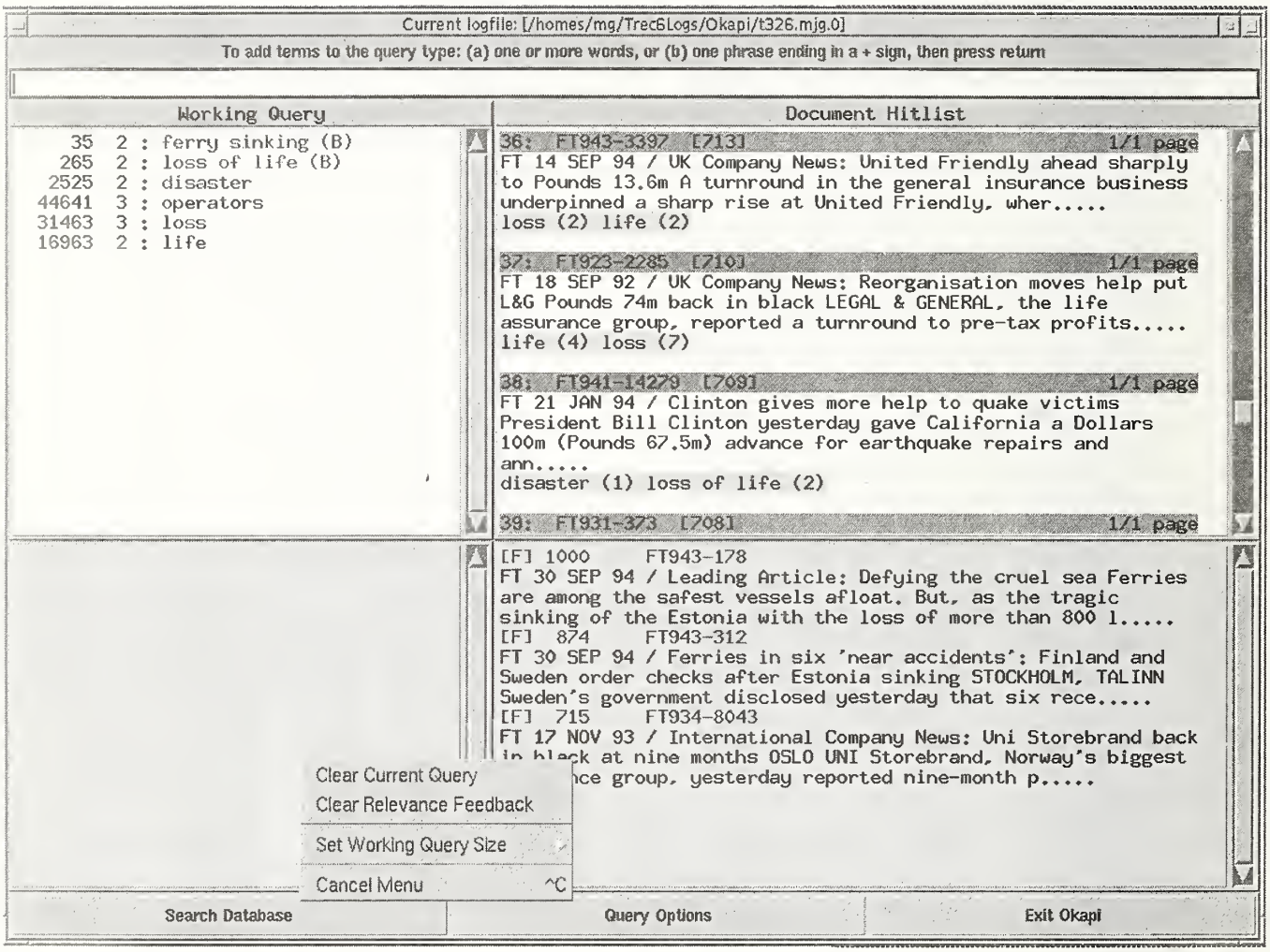


Figure 7: Interactive interface: Main Search Screen

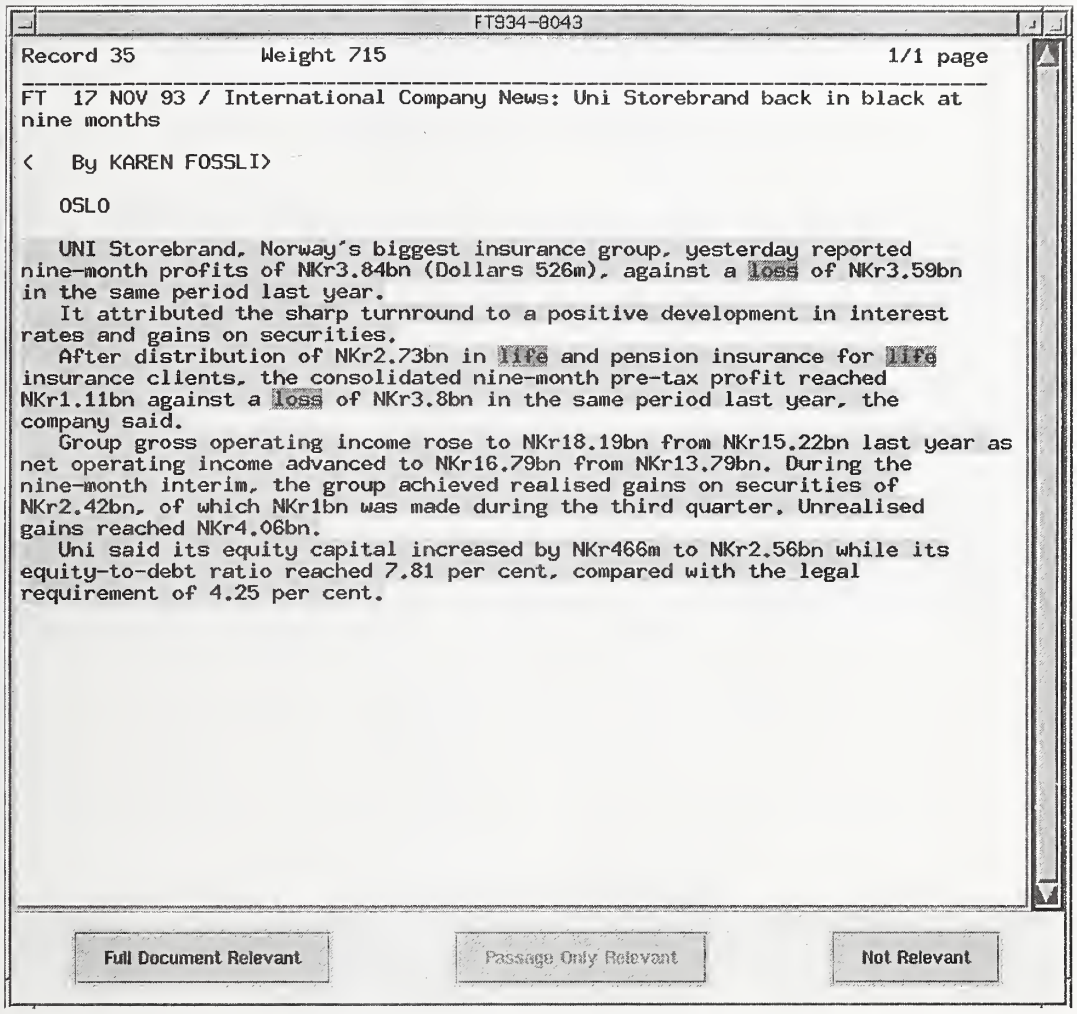


Figure 8: Interactive interface: Full Record Display

INQUERY Does Battle With TREC-6

James Allan, Jamie Callan, W. Bruce Croft,
Lisa Ballesteros, Don Byrd, Russell Swan, Jinxi Xu

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, Massachusetts USA

This year the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts participated in eight of the ten tracks that were part of the TREC-6 workshop. We started with the two required tracks, ad-hoc and routing, but then included VLC, Filtering, Chinese, Cross-language, SDR, and Interactive. We omitted NLP and High Precision for want of time and energy.

With so many tracks involved, it is nearly inevitable that something will go wrong. Despite our best efforts at verifying all aspects of each track—before, during, and after the experiments—we once again made mistakes that were minor in scope, but major in consequence. Those mistakes affected our results in Ad-hoc and Routing, as well as the dependent tracks of VLC and Filtering. The details of the mistakes are presented in each track's discussion, along with information comparing the submitted runs to the corrected runs. Unfortunately, those corrected runs are not included in TREC-6 summary information.

This remainder of this report covers our approach to each of the tracks as well as some experimental results and analysis. We start with an overview of the major tools that were used across all tracks. The paper is divided into the following sections. The track descriptions are generally broken into approach, results, and analysis sections, though some tracks require a different description.

1. Tools applied (Inquery, InRoute, LCA)
2. Ad-hoc track
3. Routing track
4. Very Large Corpus (VLC) track
5. Filtering track
6. Chinese track
7. Cross-language IR (CLIR) track
8. Spoken document retrieval (SDR) track
9. Interactive track

A. CLIR track questionnaire

B. TREC interactive track protocol log

1 Tools applied

Although UMass used a wide range of tools, from Unix shell scripts, to PC spreadsheets, three major tools were applied across almost all tracks: the Inquery search engine, the InRoute filtering engine, and a a query expansion technique known as LCA. This section provides a brief overview of each of those so that the discussion does not have to repeated for each track.

1.1 Inquery

All tracks other than the filtering track used Inquery[9] as the search engine, sometimes for training, and always for generating the final ranked lists for the test. We used Inquery V3.1 or V3.2. The former is the most recent version of Inquery made available by the CIIR; the latter is an in-house development version. The differences between the two are not consequential for this study.

The current belief function used by Inquery to calculate the belief in term t within document d is:

$$w_{t,d} = 0.4 + 0.6 \times \frac{\mathrm{tf}_{t,d}}{\mathrm{tf}_{t,d} + 0.5 + 1.5\frac{\mathrm{length}(d)}{\mathrm{avg\ len}}} \times \frac{\log \frac{N+0.5}{n_t}}{\log N + 1}$$

where n_t is the number of documents containing term t, N is the number of documents in the collection, "avg len" is the average length (in words) of documents in the collection, length(d) is the length (in words) of document d, and $\mathrm{tf}_{t,d}$ is the number of times term t occurs in document d.

1.2 InRoute

InRoute is a varient of Inquery modified to be more efficient for processing large numbers of queries on a stream of documents [8]. As a filtering engine, it processes the incoming documents one at a time. It does not have access to statistics about the incoming collection, but can use a retrospective collection for any statistics needed. InRoute has the ability to learn collection statistics as documents stream by, and can also use relevence judgements to refine a query incrementally as the training documents arrive.

Inroute was used only in the filtering track.

1.3 Local Context Analysis (LCA)

In SIGIR '96, the CIIR presented a new query expansion technique that worked more reliably than previous "pseudo relevance feedback" methods.[13] That technique, Local Context Analysis (LCA), locates expansion terms in top-ranked passages, uses phrases as well as terms for expansion features, and weights the features in a way intended to boost the expected value of features that regularly occur near the query terms.

LCA has several parameters that affect its results. The first is the choice of LCA database: the collection from which the top ranked passages are extracted. This database could be the test collection itself, but is often another (perhaps larger) collection that it is hoped will broaden the set of likely expansion terms. In the discussion below, if the LCA database is not the test collection itself, we identify what collection was used.

LCA's other two parameters are the number of top passages used for expansion, and the number of expansion features added to the query. In all cases, the LCA features were put into a query construct that allows a weighted average of the features. Assuming n features, f_1 through f_n, they are combined as:

$$\begin{array}{lcl} \texttt{\#wsum(1.0} & 1.0 & f_1 \\ & \vdots & \vdots \\ & 1-(i-1)*0.9/s & f_i \\ & \vdots & \vdots \\ & 1-(n-1)0.9/s & f_n \texttt{)} \end{array}$$

Here, s is scaling factor that is usually equal to n. The weighted average of expansion features is combined with the original query as follows:

#wsum(1.0 1.0 original-query w_{lca} lca-wsum)

where w_{lca} is the weight that the LCA features are given compared to the original query. Note that the final query is a weighted combination of the original query and the expansion features.

2 Ad-hoc track

The focus of the research carried out for the adhoc track was on query processing, query expansion, weighting and core concept identification. Most of this work was expected to produce incremental improvements compared to the techniques used in previous years, although the core concept research continues a new direction in the use of the Bayesian net model.

The official results in the ad-hoc track are significantly lower than they should be because of a failure to index Volume 5 of the test data.

2.1 Ad-hoc approach

In the query processing area, the emphasis was to produce a simpler, but effective process to replace the rather complex mixture of linguistic and statistical techniques that had been developed for TREC in previous years. The three steps in the new process are removing "stop structure", identifying phrases and proper nouns, and recognizing the presence of foreign country requirements. Stop structure refers to language constructs that are often found in queries as "fillers" and which can have occasional negative effects on retrieval. Examples of such structure are "give me documents on...", "pros and cons of...", "a relevant document will contain...", and "I am interested in...". Stop structure removal uses a table of such structures, and this part of query processing was only a minor modification of the previous year's process.

Phrase identification this year was based primarily on a phrase dictionary, rather than the part of speech tagging that was used previously. To construct this table, a lexical acquisition program was created to process large amounts of text and select suitable phrase candidates. Both part of speech and statistical approaches to identifying phrases were used, but our evaluations shows that the statistical approach was both faster and more accurate. The statistical approach, which is very similar to the statistical phrases first used by Salton in the 1970s, records phrase candidates, refines them, and then removes those with low frequencies. The phrase candidates are sequences of non-stop words, where stop words include the usual small list of words used in many retrieval systems plus irregular verbs, numbers (with some exceptions), dates, some punctuations, title words, company designators and locations. Long sequences of words are then split using rules that look for certain endings, case changes, conjunctions, and hyphenations. A final refinement checks to see if subsequences can replace longer sequences. The phrase table is then used at query processing time to identify all possible phrases in the query. Phrases are represented using the INQUERY model which decides how significant the proximity component of the phrase is and also looks for phrase words to occur in passages. This is represented as `#passage25 ( #phrase( words ))`.

For query expansion this year, we investigated refinements of the Local Context Analysis (LCA) approach first used in TREC last year and described in a recent SIGIR paper.[13] In particular, we have used different parameters for number of text passages used in the expansion and the number of concepts added to the query. In TREC-5, we found that using fewer passages (the top 20) for expansion produced better results. This was not something we observed with any other combination of database or queries. In fact, the expansion results in other tests were consistent with many more passages and 100 were used as a default in TREC-5. Although the TREC-5 queries may be unusual, we decided to be more conservative and use 30 passages this year. We also reduced the number of expansion concepts from 70 to 50. The value of w_{lca} was 1.25, meaning that the expansion features were given 125% the weight of the constructed query.

A more significant change in the LCA approach used this year was to base the expansion on passages retrieved from a larger database than just volumes 4 and 5—we used TREC volumes 1 through 5, with the Federal Register data omitted. The reason for this is simple: increasing the size of the database increases the likelihood that topical material will be retrieved and therefore increases the likelihood of finding good expansion concepts. There are two ways that this approach could negatively affect results. One is that many documents with content of little interest but containing a number of query terms could be introduced by

using the larger database. Federal Register documents are a good example of such documents. In these experiments, we excluded Federal Register documents from the archive used for expansion. The other way in which a larger database could lower effectiveness is by producing documents that, although on the correct general topic, are from the wrong time period. An example would be looking for recent documents about cooperation between Iran and Iraq, but basing the expansion on documents describing the various Iran-Iraq conflicts in the last decade. This is a problem even if just volumes 4 and 5 were used, since some of the TREC queries refer to events that are more recent than any of the data. For this reason, we did not try to correct this problem by, for example, using only documents with recent dates in the expansion.

In the weighting and core concept area, we investigated a combination of weighting and clustering techniques to identify the most important concepts in a query, including both the original concepts and expansion concepts. The process used was to weight the original query words and phrases using a combination of idf and the average term frequency in the collection. This weighting method appears to give quite reliable rankings of the importance of the concept. The weight itself, however, does not produce effectiveness improvements. Instead, we simply gave the highest ranking word or phrase a higher weight (1.5) than the rest of the query. If a single word was at the top rank, we also assigned any phrase that contained the word the same higher weight. This was intended to give the core word more context from the query. One other weighting heuristic used was that if our recognizer identified the presence of a foreign country reference in the query (`#foreigncountry`), this term was assigned the higher weight. We did this to reflect the importance of these references in many of the TREC queries.

We also looked at changing the weighting of query and expansion concepts based on how they clustered. The clustering can be based on how concepts co-occur in the collection or on how they co-occur in the retrieved documents. Although this technique shows some promise, we were not able to identify a consistently reliable implementation in time for the TREC runs. We continue to look at this issue and are also looking at using more sophisticated INQUERY operators[11] to capture models of core concepts.

2.2 Ad-hoc results

Our TREC-6 ad-hoc submissions were both flawed in that they were run against only TREC Volume 4 and not Volume 5. The following discusses the results of the *corrected* runs, not the official runs. For comparison, we include the flawed runs in Table 1 and Figure 1.

The CIIR's ad-hoc query processing included three major steps:

1. Basic query processing—removing stop phrases and stop word from the description field (for INQ401) or the title and description fields (for INQ402).
2. Phrase identification.
3. Adding up to 50 features via query expansion with LCA.

For this analysis, we applied those steps to three queries: the title, the description, and a combination of the title and description (no phrase identification was done to the title-only run). Table 1 shows evaluation numbers for the nine combinations. In all cases, each successive stage of processing improves the quality of retrieval. The very short title queries out-performed the description queries almost uniformly, but their combination provided even better retrieval quality. Figure 1 shows a recall/precision graph of the three runs (the runs represented in the bottom row of Table 1).

For comparison, the average precision for the submitted INQ401 was 0.1440, a 38% drop in effectiveness because of omitting half the collection. For INQ402's submitted run the average precision was 0.1612, a 40% drop. TREC volume 4 contains 293,710 documents, compared to the 556,077 in volumes 4 and 5, so we accidentally omitted 47% of the test collection. Of the 4611 relevant documents possible for the ad-hoc track, 58% of them came from volume 5. It is intriguing that losing 47% of the collection and 58% of the relevant documents did not cause an entirely proportional drop in effectiveness.

2.3 Ad-hoc analysis

The evaluation of the ad-hoc process by component steps as illustrated in Table 1 shows that each of the components provided some value. The identification of phrases showed a modest improvement of 4-6%,

	Title	Desc (INQ401)	Title&Desc (INQ402)	Flawed INQ401	Flawed INQ402
Basic	0.2054	0.1663	0.2103		
@20	0.3320	0.2910	0.3620		
R-prec	0.2474	0.2140	0.2461		
+ phrases	0.2149	0.1937	0.2441		
@20	0.3300	0.3240	0.3790		
R-prec	0.2668	0.2345	0.2822		
+ LCA	0.2477	0.2327	0.2730	0.1446	0.1612
@20	0.3710	0.3850	0.4200	0.2620	
R-prec	0.2910	0.2817	0.3021	0.1839	

Table 1: Comparison of three phases of ad-hoc query processing on three types of starting queries. Each cell contains the average precision, the precision at 20 documents retrieved, and the R-precision, in that order from top to bottom. The last two columns contain information about the official (flawed) runs.

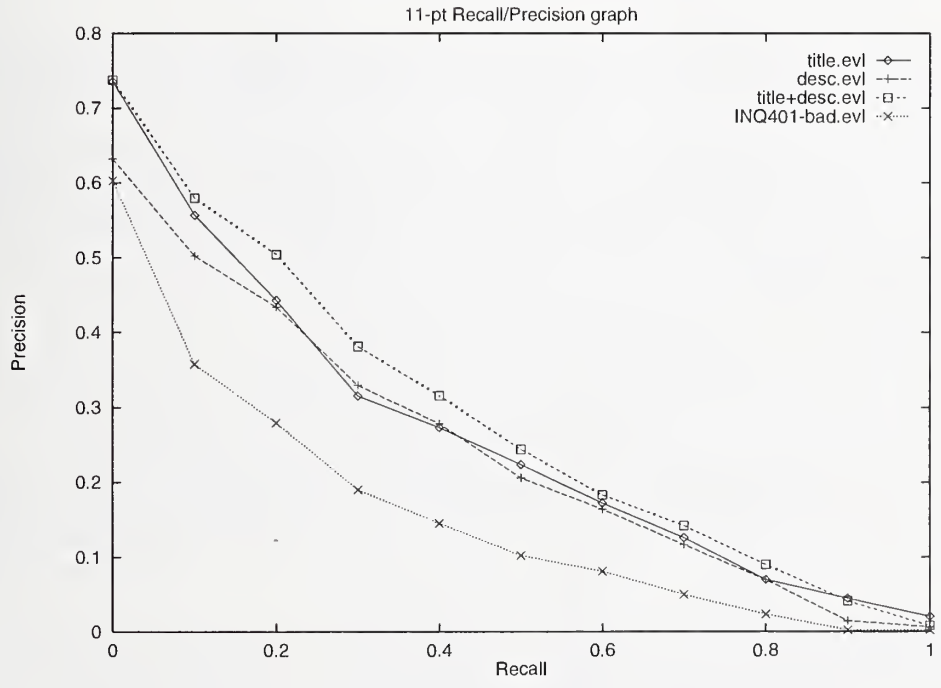


Figure 1: Recall/precision tradeoff for ad-hoc process applied to titles, descriptions, and the combination. The last two are official runs INQ401 and INQ402, respectively. (These submitted but flawed INQ401 results are provided for comparison.)

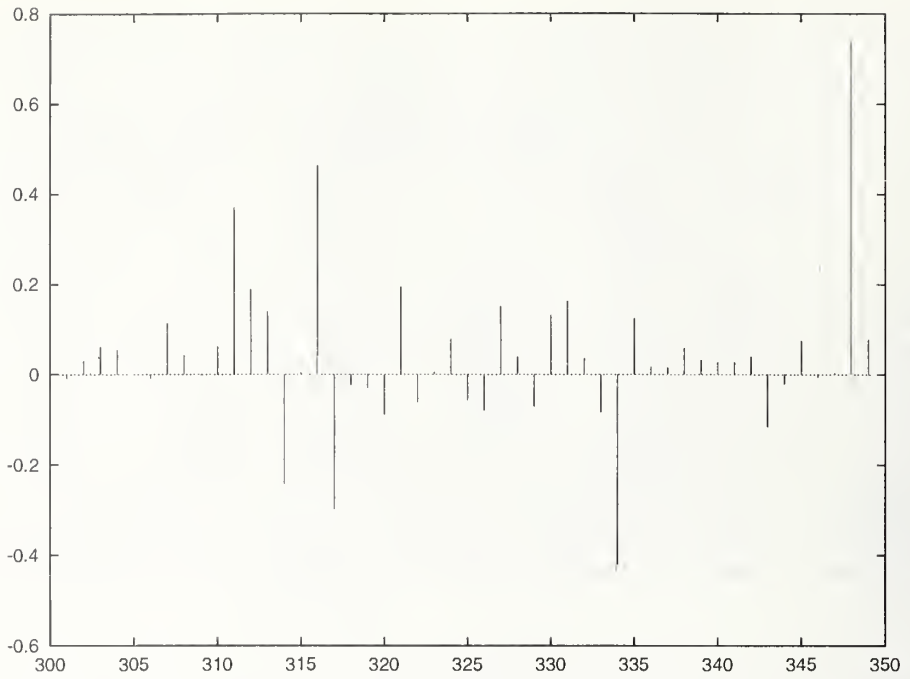


Figure 2: Change in average precision for the ad-hoc queries when the title is used as a basis for the query rather than the description. These results are for basic query processing.

though it is not statistically significant (by a sign test). The additional 15% or so improvement caused by the LCA expansion is, however, statistically significant at virtually all levels of recall and all document cutoff values.

One of the more interesting characteristics of the queries is the noticeably better effectiveness that the short, 2 or 3 word title queries achieve as compared to the longer descriptions. The difference is almost entirely wiped out by our query processing, but it remains even then. A sign test shows that the difference is statistically significant, with a P-value of 0.0325, but it is only the average precision that is significant: the difference is not significant at any standard recall point other than 0.0, nor at document cutoffs of 5, 10,15, 20, 30,100, 200, 500, or 1000.

Some quick scanning of the results shows that although most of the title queries are substantially better, there are some that are not. Figure 2 shows the difference in average precision for the queries when the titles are used rather than the descriptions (a positive number means the title query is better). The startling quality of the very short queries is not particularly surprising considering the following:

- Topic 349 is about metabolism (it showed the greatest change by using titles). The title query is "metabolism". The description provides a definition of metabolism without using the word.
- Topic 316 is about polygamy. The title is very specific. The description includes noise words that will confuse most query engines: roots, prevalence, world, today.
- Topic 311 is about industrial espionage, but the the description mentions neither industry or espionage.
- Topic 312 is about hydroponics, but the description does not mention hydroponics.

We have not investigated whether the odd query construction in fact caused any of the mistakes in the system (perhaps articles about hydroponics only occasionally mention "hydroponics"), but it seems to be the root issue in many cases.

The LCA query expansion appears to have helped in most of those cases: Topic 311 is expanded to include "espionage", 312 gains "hydroponics", 316 now includes so many references to polygamy that the

noise words are lost. Topic 349 is not helped by expansion, perhaps because it fails to acquire the word "metabolism."

3 Routing track

UMass had very little research interest in the routing track this year, and unfortunately that appears to have shown in the results: a careless error in the query running caused a large number of query terms to be entirely ignored. The approach for query formulation was very similar to that taken in TREC-5, with some minor exceptions.

3.1 Routing approach

The basic approach to the routing task was similar to last year's method. The query is exanded by extracting features that occur often in the relevant documents and rarely in the non-relevant document. Feature weights are assigned as a Rocchio combination of weights in the relevant and non-relevant documents. The final weights are adjusted using Dynamic Feedback Optimization.[6]. The peculiarities of this year's approach are as follows:

- A starting query Q_0 was created from the *all* parts of the routing topic using the methods described in the ad-hoc track.
- In TREC-5, we built 8 different training databases for the 50 routing queries. Those databases represented all possible combinations of the TREC volumes on which a routing query had been evaluated in the past. The result was that when a query was run against its training database, any unjudged documents are highly likely to be non-relevant, since that database had been at least partially judged.

  For TREC-6, we made an effort to reduce that work substantially. We built one extremely large database that included TREC volumes 1 through 4, as well as the TREC-4 and TREC-5 routing volumes (there is some overlap in those volumes; documents were not indexed twice). The training documents were selected by running Q_0 against the training database and then removing any documents that were not explicitly judged (i.e., were not in the TREC relevance judgements list), resulting in the training set S_0. A second run of Q_0 retrieved the top-ranked 200-word passages in the training collection, similarly restricted to passages from judged documents, yielding P_0.
- The documents in training set S_0 were examined and all terms that were not stop words were extracted. In addition, any phrases that occurred in the set of phrases used for ad-hoc query construction were also extracted. The result was a list of words and statistically common phrases occurring in the training documents. The training passages in P_0 were also examined for all pairs of words that occurred within a window of 20 of each other inside the passages.

  The words and phrases were sorted by the proportion of relevant training documents containing the feature minus the proportion of non-relevant training documents containing it. A feature that occurred in all of the relevant documents and no non-relevant documents would have a weight of 1.0; a feature that occurred evenly in both sets would have a weight of 0.0; and so on. The 20-window words were similarly ranked.
- A query was constructed from the features of the original query, the 20 most highly weighted terms, the 20 most highly weightedphrases, and the 20 most highly weighted 20-window pairs, for a total of up to 60 features added. In no case was a feature added if its weight from above was below 0.045.

  The features were all assigned the weight:

$$w_q + 4w_r - \frac{1}{2}w_{nr}$$

  where w_q was the weight in the original query (zero if the feature was not in the query), w_r was the average tf value of the feature in the relevant documents (*not* the average belief), and w_{nr} was the

	INQ403 (correct)	INQ403 (submitted)
Avg prec	0.3180	0.2290
Prec @ 20	0.5106	0.4617
R-prec	0.3576	0.2898

Table 2: Routing results, showing both a correct run as well as the results from the submitted run that had large amounts of the query ignored.

average tf value of the feature in the non-relevant documents (zero if the feature did not occur in the non-relevant documents).

This created query Q_1.

- Query Q_1 was run against the training collection again and all judged documents in the top 20,000 retrieved documents were used as the basis for DFO adjustment of the weights. DFO was applied in three passes, allowing the weights to increase by 100%, by 50%, and by 25%, respectively. The resulting query is Q_2.
- Q_2 was the final query submitted to NIST and run against the test collection.

The differences between TREC-5 and TREC-6 are that an *a priori* set of statistical phrases was used rather than mining the training set for common pairs of adjacent words, for pairs within a window of 5, and for pairs within a window of 50. Further, in TREC-5 the queries were expanded with up to 250 features whereas for TREC-6 we allowed only up to 60 additions.

3.2 Routing results

Unfortunately, the process of gathering retrospective statistics for various idf values of features contained a bug. The result was that large numbers of query features were treated as if they did not occur in the database—e.g., for topic 1, 46 of 118 features were dropped from the query, resulting in a 25% drop in average precision (for that topic).

Table 2 and Figure 3 show the results of the routing run. In both cases, the submitted run is included along with the corrected run for comparison. (The 25-40% improvement from the bad run to the good run is statistically significant at all levels after the top 15 documents are retrieved.)

3.3 Routing analysis

Beyond error analysis to determine why the results were so bad, no work has been done at this time to understand how the routing query formulation worked.

4 Very Large Corpus (VLC) track

Our goal for the Very Large Corpus (VLC) track was to build and search a single database of 20 gigabytes (GB). Inquery had been tested elsewhere on databases of comparable size, so we did not expect size to be a problem. We were interested primarily in studying the times required to index and retrieve documents from a 20 GB database.

4.1 VLC approach

The indices were built in two stages. In the first stage, during document parsing, a series of temporary files were written that each contained one or more blocks. Each block was a set of inverted list fragments. When all document files had been parsed, the second stage began. In the second stage, temporary files were merged, yielding a final inverted index.

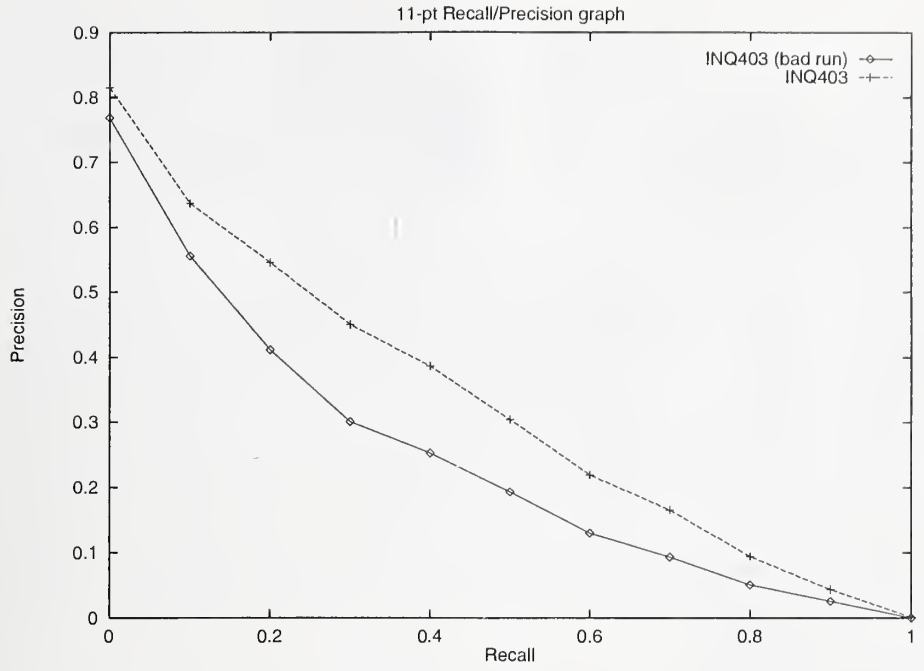


Figure 3: Recall/precision graph for INQ403, the routing run. Both the submitted and correct runs are shown for comparison.

The times required to build the 2 GB Baseline and the 20 GB Full VLC database are shown below. The figures do not include the time required to copy files from CD-ROM or DAT tape, nor the time required to uncompress the files. The experiments were run on an UltraSparc computer with 4 processors and 1 GB of memory, primarily because that machine had the (ample) disk space required for indexing the VLC corpus. Only one processor and less than 100 MB of memory were used.

Task	2 GB Time (hh:mm:ss)	20 GB Time (hh:mm:ss)	% CPU
Parse	5:55:29	61:21:16	97%
Merge	36:51	4:40:53	71%
Total	6:32:20	66:02:09	

The 2 GB index was built at a rate of 308 MB per hour, while the 20 GB index was built at a rate of 303 MB per hour. It is encouraging that indexing time scaled linearly. However 300 MB per hour is slower than expected, so we view these figures with caution.

Instead of creating new queries for the VLC track, we used the queries created for the ad-hoc track (see Section 2).

4.2 VLC results

Timing and accuracy figures are shown below for two official and four unofficial runs. The timing figures were obtained after "warming-up" the system by running query 251 from the INQ301 query set used in TREC-5. Each query returned 20 documents, as specified in VLC track guidelines.

		Full Index			Top-Docs, 1K		Top-Docs-Only, 1K	
Database	Query Set	Run ID	Time per qry (m:ss)	Prec 20	Time per qry (m:ss)	Prec 20	Time per qry (m:ss)	Prec 20
2 GB	INQ402	INQ414	0:41	0.387	0:23	0.389	0:20	0.324
20 GB	INQ402	INQ412	6:50	0.505	3:48	0.497	2:59	0.332

4.3 VLC analysis

The most striking result of the VLC experiments is that precision is far higher on the 20 GB corpus than on the 2 GB baseline corpus. This result is not unique to Inquery; every group participating in the VLC track had similar results. Its cause is unknown, although it may simply be that the larger database had more relevant documents.

A second result was that query time scaled linearly with the size of the database. This result was expected, because we used a version of Inquery that does not do any form of optimization.

An unofficial experiment tested the effects of *top-docs* optimization, in which each query term contributes only its best 1,000 documents to the ranked list. The top-docs optimization had minimal impact on precision while doubling the speed of document retrieval, which is consistent with published results [5].

Another unofficial experiment tested the effects of *top-docs-only* optimization, in which each query term contributes a score for only its best 1,000 documents. The top-docs-only optimization improved speed by another 13-21% (as compared with the top-docs optimization), but reduced precision by 17-33%. These results were a surprise; we expected more of an improvement in speed, and less of a loss in precision.

The timing experiments demonstrate that the current optimization techniques do not provide the speed necessary to run highly complex queries on a 20 GB database. The queries created for TREC Ad-hoc experiments contain an average of 99 terms and 31 query operators (primarily proximity and phrase operators) per query. Although effective, few people would wait 3-4 minutes for query results – even for very good results. A combination of more concise queries and improved optimization techniques are required for very large corpora.

5 Filtering track

Our goals for the Filtering track were to use InRoute, our document filtering system [8], for all of the experiments, and to use an incremental Rocchio algorithm [1] for the Adaptive Filtering experiments. These were modest goals, given our previous work. The only new work required was an algorithm to learn dissemination thresholds incrementally.

5.1 Filtering approach

The "batch-learned" experiments were of minimal interest to our group, because of their similarity to the Routing track. For example, the batch-learned profiles in all of our Filtering experiments were created with the same techniques used in the Routing track (described above). The filtering experiments merely used a more restricted set of corpus statistics and relevance judgements. The batch-learned dissemination thresholds were the "optimal" thresholds for the training data [2].

Our interest in the "batch-learned" experiments was confined to seeing the effects of different corpus statistics, and the effects of different evaluation metrics. Consequently, seven of our ten runs are quite similar.

The Adaptive Filtering experiments were the most interesting to us because of their similarity to "real world" environments. Each topic was converted automatically into an AdHoc query, using a subset of the techniques used in the AdHoc track (described above). The initial dissemination threshold was set low enough that matching on any query term would exceed the threshold.

During the training phase, if a document was selected for dissemination, InRoute was given that document's relevance judgement; unjudged documents were treated as not relevant. Profiles were modified using

Run ID	Profile Method	Threshold Method	Corpus Stats	Metric	Prec100	AvPrec
INQ415	Batch	Batch	FBIS 3,4	F1	0.1111	0.0499
INQ416	Batch	Batch	FBIS 3,4	F2	0.1705	0.0734
INQ417	Batch	Batch	TREC 1,2,3 +	F1	0.0746	0.0391
INQ418	Batch	Batch	TREC 1,2,3 +	F2	0.1417	0.0656
INQ419	Batch	Batch	FBIS 3,4	ASP	0.0087	0.0039
INQ420	Batch	Batch	TREC 1,2,3 +	ASP	0.0115	0.0046
INQ421	Online	Online	FBIS 3,4	N/A	0.2670	0.1683
INQ421c					0.3297	0.2074
INQ422	Online	Online	TREC 1,2,3 +	N/A	0.2924	0.1698
INQ422c					0.2817	0.1794
INQ423	Online	N/A	FBIS 3,4	Ranked	0.2668	0.2067
INQ423c					0.3270	0.2774
INQ424	Batch	N/A	FBIS 3,4	Ranked	0.2306	0.1525
INQ424c					0.2864	0.2075

Figure 4: Summary of the ten UMass Filtering runs. Run names postfixed with "c" are corrected versions of the official TREC submissions.

an incremental Rocchio algorithm [1]. Thresholds were modified to be halfway between the average relevant document score and the average nonrelevant document score.

Profiles and thresholds were "frozen" during the testing phase.

The three adaptive runs differ in the corpus statistics used, and the way in which they are evaluated.

Although 10 runs were submitted (Figure 4), the number of ideas tested was small.

- INQ415, INQ416, and INQ419 are identical *except* threshold learning; thresholds in these runs were "optimized" for different evaluation metrics (F1, F2, and ASP, respectively).
- INQ417, INQ418, and INQ420 are the same as INQ415, INQ416, and INQ419 *except* that a broader set of corpus statistics was used during filtering (TREC 1,2,3,+ instead of FBIS 3,4).
- INQ422 is the same as INQ421 *except* that a broader set of corpus statistics was used during filtering (TREC 1,2,3,+ instead of FBIS 3,4).
- INQ423 and INQ424 are the same as INQ421 and INQ415, but are evaluated as ranked runs.

The same "batch learned" profiles were used for runs INQ415 – INQ420, and INQ424; only the thresholds and corpus statistics differed among these runs. The "batch learned" profiles were learned using only FBIS 3 and FBIS 4 training data and corpus statistics.

5.2 Filtering results

The results are summarized in Figure 4. Several small errors were found and fixed after the official submissions, which led to improved results. The corrected runs are also shown in the table, with the suffix "c" added to the original run id.

5.3 Filtering analysis

Most of the batch-learned-profile experiments (INQ415-INQ420) produced poor results, due to poor selection of batch-learned thresholds. For example, the median number of documents disseminated by experiment INQ415 was 4. We have not yet done failure analysis to determine what caused the batch-learned thresholds to be so poor.

The one experiment that evaluted batch-learned-profiles using ranked retrieval (INQ424), instead of a dissemination threshold approach, produced results that were similar to Routing track experiments. This

result was expected, because the experiment was essentially a Routing track experiment; the only differences for the Filtering track were a narrower set of corpus statistics (as required), and less accurate *idf* values for proximity operators (corrected in run INQ424c).

The experiments that tested adaptive learning methods were far more encouraging. Profiles learned adaptively (INQ421c-INQ423c) had precision and recall comparable to profiles learned with a batch method (INQ424c). Recall was lower when adaptively learned thresholds were applied (compare INQ421c to INQ423c), however the difference was smaller than expected (almost *any* threshold lowers recall). In these experiments, the adaptive methods of learning profiles and dissemination thresholds were quite effective.

Analysis of the experimental results is complicated by the fact that INQ421c and INQ422c did not find documents for every query, due to their use of thresholds. INQ421c found documents for only 36 queries, while INQ422c found documents for 46 queries. This result suggests that a broad set of corpus statistics is more effective than a narrow set, but one cannot draw strong conclusions from this one comparison.

It is difficult to draw conclusions about the threshold algorithm from these results. Clearly it did not cause profile-learning, Precision at 100 documents, or Average Precision to deteriorate substantially. However, a threshold of 0 produces similar results when there are many relevant documents. Further work is required to determine the effectiveness of the threshold algorithm.

Although many questions remain, the results from these experiments are encouraging. The adaptive results are comparable to batch results, even though proximity operators are not yet included in adaptive queries. (Proximity operators normally improve effectiveness significantly.) This result, by itself, is cause for optimism.

6 Chinese track

For TREC-6, we did not attempt any new processing of the queries or database for the Chinese track.

6.1 Chinese approach

The Chinese retrieval experiments are similar to the work done for TREC-5.

1. To allow for flexibility in segmentation at query time, each Chinese character is indexed as a term. Exceptions are made for characters making up numbers and the elements of dates which are indexed as a group.

2. Queries are made up of the title and description fields of the topics. They are automatically preprocessed to remove punctuation. These basic queries are then automatically segmented using the USEG segmenter, based upon hidden Markov models. Each segmented Chinese word is represented by a proximity operator which requires that the glyphs be immediately adjacent and in order. To compensate for possible segmenter errors, sequences of single characters are wrapped in a `#phrase` operator with the restriction that all glyphs be within a window of 25 terms. Each word in the description is weighted as a single term (weight 1.0) while isolated single terms are downweighted (weight 0.3). The whole title is weighted as a single term (weight 1.0).

3. The queries are expanded using Local Context Analysis (LCA). The basic query is used to retrieve the top-ranked passages for each topic. LCA is applied to extract expansion words from the top-ranked passages. An expansion word is a segmented word as defined by USEG. The segmenter is augmented with a name recognizer to reduce errors of name segmentation. The top 70 words from the top-ranked passages are added to the query. Each concept is assigned a weight in decreasing order. Word_i is assigned the weight $w_i = 1.0 - 0.9(i-1)/70$. Two runs are done. The first, INQ4ch1, extracts the expansion words from the 10 top-ranked passages retrieved and the second, INQ4ch2, from the 20 top-ranked passages retrieved. The expansion section of the final query is given twice the weight of the original query.

6.2 Chinese results

The following table summarizes our Chinese runs.

	Avg Prec	Prec @ 20	R-prec
title+desc, noseg	0.4785	0.7288	0.4831
title+desc, seg	0.4554	0.6827	0.4665
desc+seg	0.4209	0.6538	0.4324
title+seg	0.3743	0.5788	0.4088
INQ4ch1	0.5336	0.7654	0.5218
INQ4ch2	0.5223	0.7538	0.5137

6.3 Chinese analysis

It is surprising that the segmentation actually hurts the queries. We have not yet examined why this is true.

7 Cross-language IR (CLIR) track

The cross-language retrieval experiments focused on disambiguating translations of Spanish (source) queries to English (target). A parallel corpus of UN documents from 1988-1990, obtained from the LDC, was used in addition to POS tagging to disambiguate term translations. Phrases were translated via information extracted from the Collins Spanish-English machine readable dictionary (MRD). Local Context Analysis (LCA) was employed prior to and after query translation to reduce the effect of poor translations.

A more detailed discussion of some of the techniques used in this track was published recently.[4] Appendix A includes the CLIR Track Questionnaire.

7.1 CLIR approach

Query processing for the cross-language experiments begins with part-of-speech (POS) tagging using the MITRE POS tagger. As is the case with English queries, stop phrases are removed. With the exception of adjacent proper nouns which are treated as phrases, query and expansion terms terms in the source language are translated to the target language using the Collins MRD. The term translations are then disambiguated with the UN corpus. A more detailed description of query translation follows.

Each tagged query term is replaced with the source language equivalent term or terms that correspond to its part-of-speech. If there is no translation corresponding to a particular query term's tag, the translations for all parts-of-speech listed in the dictionary for that term are returned. There may be one or more ways to translate a given term. When more than one equivalent is returned, the best single term is chosen from this list via parallel corpus disambiguation.

Disambiguation proceeds in the following way. The top 100 Spanish documents are retrieved from the parallel UN corpus using the original Spanish query. The top 5000 terms based on Roccio ranking are extracted from the English UN documents that correspond to the top 100 Spanish documents. The translations of a query term are ranked by their weight in the list of 5000. The highest ranking equivalent is chosen as the "best" translation for that term. If more than one translation equivalent have the same rank, they are all chosen. If none of the equivalents are on the list, no disambiguation is performed and all equivalents are chosen.

Phrasal translations were performed using information on phrases and word usage contained in the Collins MRD. This allowed the replacement of a source phrase with its multi-term representation in the target language. When a phrase could not be defined using this information, it was translated word-by-word as described above.

Translated queries are then expanded using Local Context Analysis. When expanding, the top 50 concepts were added from the top 30 passages with multi-term concepts wrapped in the INQUERY #phrase operator with the restriction that all terms be found within a window of 25 terms. For example, #passage25(#phrase(president kurt waldheim)). Concepts were weighted with an infinder-like weighting scheme. The top concept

was given a weight of 1.0 with all subsequent concepts down-weighted by $\frac{T-i-1}{T}$, where T is the total number of concepts and i is the rank of the current concept.

Two sets of queries were generated, one using only topic descriptions (INQxl2) and the other using both descriptions and titles (INQxl1). The original query translation and additional concepts were combined as described in the discussion of LCA (Section 1.3) with w_{lca} set to 1.0.

7.2 CLIR results

Two sets of results, INQxl1 and INQxl2, were submitted in the Cross-language track. Both sets were based on automatic processing of TREC topics CL1-CL25 into queries and automatic query expansion. The official results for 21 queries are summarized below. Table 3 compares effectiveness of English queries consisting of title plus description with queries INQxl1. Table 4 compares effectiveness of English description only queries with queries INQxl2. In both cases, the baseline English queries were expanded with the top 50 concepts from the top 30 documents.

Query Type	Precision			
	5 docs	30 docs	100 docs	Avg Prec (NI)
Desc	0.5429	0.4683	0.2814	0.3721
INQxl2	0.2095	0.1825	0.1167	0.1810 (-51.4)
INQxl2-fix	0.4000	0.3095	0.2043	0.2528 (-32.1)

Table 3: Results for title and description queries.

Query Type	Precision			
	At 5 docs	At 30 docs	At 100 docs	Ave Prec (NI)
Desc+Title	0.6000	0.4905	0.3081	0.4113
INQxl1	0.3048	0.2778	0.2010	0.2610 (-36.5)
INQxl1-fix	0.3619	0.3095	0.2019	0.2593 (-36.9)

Table 4: Results for description only queries.

Early analysis revealed programming errors which led to key query term translations being eliminated. For example, the pre-translation expansion term translations were not included in any query. We re-ran these experiments after eliminating the errors and the are shown in the third row of tables 3 and 4.

7.3 CLIR analysis

In the absence of complete relevance judgments, we are unable to perform an accurate analysis. However, we can say how these results compare to earlier work in cross-language retrieval. Cross-language retrieval via simple dictionary query translations [4, 3, 10, 12] tends to yield effectiveness which is 40-50% of monolingual retrieval effectiveness. Our cross-language description only query (INQxl2) results are consistent with this. Dictionary translations can be disambiguated via pre-translation and post-translation query expansion [4] or via part-of-speech and parallel corpus disambiguation [10], yielding cross-language effectiveness that is 70% of monolingual.

The TREC results are consistent with earlier results. However, we were surprised to find that pre-translation expansion alone was not particularly effective. We speculated that the overall effectiveness of the combined expansion method would improve if the effectiveness of the pre-translation expansion phase were improved. This turns out to be the case.

Table 5 shows representations of query 19 with both description and title. First is the original English, second the Spanish version, third the top 5 pre-translation expansion terms for the Spanish query, fourth the UN disambiguated translations of the expansion terms, and fifth the correct translations of the expansion terms. The disambiguation chooses the wrong translation about 20% of the time, shifting the query away from

the correct context. Post-translation expansion may then pull in more unrelated concepts. If disambiguation is not used for expansion term translation, effectiveness of the pre-translation expansion increases as does the effectiveness of combining pre- and post-translation expansion. Table 6 shows an increase in effectiveness to 73% of monolingual when parallel corpus disambiguation is not used on the expansion term translations. Row one shows the original INQxl1 results and row two gives results for these queries without expansion-term corpus disambiguation. It is clear that although corpus disambiguation is effective, poorly disambiguated translations can have a large negative effect on performance.

The effect of each stage of the translation process as a percentage of monolingual average precision can be seen in table 7.

English	Wine. Is wine consumption production rising or decreasing world-wide?
Spanish	Vino. Está la producción consumo de vino creciendo o decreciendo a nivel mundial?
Exp. Terms	vino vinos consumo producción hule (bad term)
Dis. trans	party party consumption production rubber
Correct trans	wine wine consumption consumption n/a

Table 5: Query CL19

Query Type	Precision			
	At 5 docs	At 30 docs	At 100 docs	Ave Prec (NI)
INQxl1	0.3619	0.3095	0.2019	0.2593
INQxl1-no_dis	0.4095	0.3730	0.2424	0.3012 (+16.1)

Table 6: Precision at low recall and average precision for INQxl1 with and without corpus disambiguation of pre-translation expansion terms.

Query	Avg. Prec	%Monolingual
WBW	0.1570	38
WBW+Phr	0.1629	40
WBW+Dis	0.2099	51
WBW+Dis+Phr	0.2551	62
WBW+Dis+Phr+Pre	0.2454	60
WBW+Dis+Phr+Post	0.2864	70
WBW+Dis+Phr+Combined	0.2864	73

Table 7: Effect of translation steps as a percentage of monolingual average precision. WBW: word by word translation; Phr: phrase (proper nouns) recognition and translation; Dis: POS and UN corpus disambiguation; Pre: pre-translation expansion; Post: post-translation expansion; Combined: pre- and post- translation expansion.

8 Spoken Document Retrieval (SDR) track

Our efforts in this track compared runs on three databases: the human transcribed text, the provided recognized text, and text recognized by Dragon Systems on our behalf. In all cases, we used minimal query processing methods and two rounds of LCA to generate the queries.

8.1 SDR approach

Our SDR work utilized four sets of documents:

1. The LTT corpus provided by NIST. These are human-transcribed texts of the audio corpus. They provide the expected upper bound of performance.
2. The IBMSRT corpus, also provided by NIST. This corpus is the result of IBM's providing speech-recognized text for use by the entire SDR group. It is degraded text.
3. The DRAGON corpus, built by Dragon Systems, our partners in this track. This corpus is also degraded text. The method used by Dragon to create the text is provided below.
4. The Topic Detection and Tracking (TDT) corpus available via the Linguistic Data Consortium. This is a set of about 16,000 news stories from Reuters and CNN, covering July, 1994, through June, 1995. It was used in this track as a reliable (non-degraded text) corpus covering a similar time period as the test corpus.

The first three were test corpora and final queries were run against them for submission to NIST. The last corpus was used only during query construction.

For each test corpus, we created a 3-part query. The parts were:

1. The original query with stop-phrases removed and phrases identified as in the ad-hoc track (Section 2).
2. An LCA expansion of the original query using the TDT corpus. Up to 29 features were added from the TDT corpus. These were intended to provide additional features from a related corpus of high quality. (LCA expansion is described in Section 1.3.)
3. An LCA expansion of the original query using the test corpus (either LTT, IBMSRT, or DRAGON). Up to 29 features were added here, too. These were intended to expand the query based on the database to provide topical vocabulary.

The three parts were combined as a weighted sum:

```
#wsum(1.0 10.0 original-query
      2.0 test-LCA
     10.0 TDT-LCA )
```

Note that the expansion features from the test corpus were down-weighted relative to the other features. This was done because we felt that features extracted from a degraded database would be less reliable.

8.2 SDR speech recognition

The speech recognition component of our TREC SDR work (labeled "DRAGON" above) was accomplished by Dragon Systems. This section describes the process they used to transform the audio into text.

8.2.1 Acoustic models

The frontend that we are using has 36 features, namely 12 modified plp cepstra (including C0), and the corresponding first and second differences. Channel normalization is done within a given speaker's data.

The phone set that we are using is larger than we have used in the past: 51 phonemes (including silence) instead of the 43 phoneme set that we have used before. It is larger because certain vowels have stressed and unstressed versions, and it includes syllabic consonants.

We trained acoustic models using the first half of the HUB4 acoustic training corpus. We only used the first half so that we could use these models in the TREC SDR evaluation. This half of the data consists of about 34 hours of usable training material—however to start with, we trained only from speakers that had a minute or more of data in the first half. Overall, 27 hours of data distributed among 417 speakers satisfied this condition.

We used gender-independent models trained from a 24 hour subset of the WSJ si284 corpus to obtain initial alignments of the HUB4 data.

8.2.2 Clustering

In the TREC SDR evaluation we did not use the speaker side information, so we needed to develop a clustering algorithm that would group the data into clusters that corresponded to the actual speaker clusters.

To do the clustering, we use a k-means algorithm that uses the following distance measure of a segment s to a cluster c:

$$\mathrm{KL}(s, c+s) + \mathrm{KL}(c, c+s) + \mathrm{TimePen}(c, s)$$

where $\mathrm{KL}(a, b)$, the Kullbeck-Leibler distance, is the expectation under a (as the true hypothesis) of the logarithm of the ratio of the probability of the a distribution to the probability of the b distribution, and $\mathrm{TimePen}(a, b)$ is a linear function of the smallest time difference between a frame in a and a frame in b, truncated at a maximum value.

8.2.3 Language model

We used an interpolated language model consisting of two components:

1. A bigram language model trained from the first half of the acoustic training transcripts (roughly 400,000 words, with all bigrams kept).
2. A trigram language model trained from 62 million words of Journal Graphics transcriptions of broadcast news sources from the period January 1995 through April 1996 (kept all bigrams, but only trigrams that occurred three or more times). The Journal Graphics transcripts were processed to covert them from "written" text to "spoken" text.

Interpolation weights were trained from the 1996 HUB4 evaluation transcripts. The 56,000-word lexicon was constructed from three sources:

1. the 18,000 distinct words found in the first half of the HUB4 training data
2. the 19,000 most common new words found in the Journal Graphics data
3. the 19,000 most common new words found in 50 million words of newspaper data taken from the 1995 Philadelphia Inquirer.

8.3 SDR results and analysis

To illustrate the query processing methods, we consider Topic 3 in the SDR track. Words in quotation marks are phrases.

- *Original*: What is the difference between the old style classic cinemas and the new styles of cinema we have today?
- *Basic query processing*: difference "old style" old style classic cinemas new styles cinema
- *TDT expansion features*: frankenstein "film industry" "kenneth branagh" cinema film fad style lowrie "paris cinema" "fred fuchs" "francis ford coppola" "cinemas benefit" "century rendition" "century horror classic" "adrian wootton" "art form" prod. "mary shelley" casting technician "thai house" "peter humi" profit "robert deniro" popularity "margaret lowrie" helena hollywood image
- *IBM-recognized text expansion features*: heart yeltsin loom dollar "style rally" "russians dozen" "men mahal room hut" "m. men" "louisville ala" lerner "house canvassing" "ham men" "election spending" "economists yeltsin" "campaign team" "attitude moon" percent "v. broadcast" "soprano maria callas" "new line cinema" monitoring "mel gibson" janine "daniel m. t." movie "lou duva" equivalent news singapore
- *Dragon-recognized text expansion features*: years trent houses emission style graduate pandering nights negotiations cinema barrels awards kidney lott enemies "years industry" sander "houses emission" "g. o. p. fire brand set" wilderness tumor melting "majority leader trent lott" literature "cover story" dennis "house republicans" toronto soprano sequence

It is clear from the expansion features that the recognized text caused expansion with very poor, generally unrelated features.

The following table lists the number of topics (out of 49) where the known relevant item was ranked first by our system, and where it was found somewhere in the top 10 (including the first rank). Note that for the two topics that had two relevant documents (43 and 48), we always found those at ranks 1 and 2.

	LTT		IBMSRT		Dragon	
	top	top10	top	top10	top	top10
Basic	38	46	33	42	36	43
+TDT	35	45	25	42	34	44
+LCA	40	46	32	42	38	43
all	39	45	32	42	38	45

In the table, the rows correspond to basic query processing, adding the TDT expansion concepts, instead adding the expansion concepts from the database in question, and adding both sets of concepts. The colums correspond to the three collections: human-transcribed, machine transcribed for NIST, and Dragon's machine transciption.

Given the apparent quality of expansion concepts from the TDT and test corpora list above, it is surprising that adding the TDT concepts consistently hurt performance and adding the others often helped. However, Topic 3 may not be the ideal sample. The following lists three spectacular failures of the system:

1. In Topic 3, the known item was retrieved at rank 27 on the human transcribed corpus, and rank 209 on the Dragon run.

2. In Topic 42 (fashion in beach coverups), the relevant document was found at rank 24 (Dragon corpus). The TDT expansion added words that were vaguely on-point, but the Dragon expansion included oil refineries, coastlines, and wildlife refuges because of the word "beach."

3. In Topic 47 (the Valujet crash), the relevant document was found at rank 36 (Dragon corpus). This is primarily because although the TDT expansion included information about the Everglades, it focused on the sugar industry.

The errors in our system appear to be primarily the result of mistakes in query expansion—i.e., expanding the wrong word or the right words but in the wrong way—rather than because of limitations in the recognition of speech.

9 Interactive track

We designed a novel interface specifically for doing aspect oriented retrieval. This system had the following features:

- In order to save a document, it was necessary to drag it to an area reserved for aspects.
- Significant terms were extracted from documents grouped into an aspect to help the user in labelling an aspect.
- Color coded visual cues were provided to show a user if a document had been viewed before or not.
- A 3-D map was given to the user where documents with high similarity were placed close together.

Because the interface to our system was quite different from the control system, ZPRISE, and because it included two distinct visualizations (discussed below), we decided that even if a significant difference was found between our system and ZPRISE we would not know which part of the interface caused that difference. We then made two versions of our system: our full system ("AspInquery Plus") contained all the features listed above, and a more basic version ("AspInquery") that only used one of the visualizations (the only change to the code was commenting out a call to the constructor for the other visualization). If a large

difference in performance was observed between the two systems we would then know what feature had caused it.

The work described below is discussed in more detail elsewhere.[7] Appendix B includes the protocol for one participant on Topic 1.

9.1 Interactive approach

As required by NIST, we ran ZPRISE as a control system; the two experimental systems were basic and extended versions of one program. The extended version ("AspInquery Plus") simply added a 3-D window to the basic system ("AspInquery") . Both versions use the well-known Inquery search engine. The core of our user interface has much in common with the ZPRISE interface, differing in two significant ways: ZPRISE displays the query terms contained in a document after the headline but our system does not, and our system color codes whether a document has been viewed but ZPRISE does not. Specifically, we write the headline information of a document in blue if it has not been viewed before, and purple if it has been seen. (This scheme was modeled after the default color scheme Web browsers use to show if a hypertext link has been followed or not.)

Both ZPRISE and our system accept plain text input for queries. Our system also supports a phrase operator, invoked by placing terms together within double quotes (e.g., "balanced budget"). The phrase operator increases the ranking assigned to documents where all terms in the phrase are found in close proximity. (In reality, our system supports the full syntax of Inquery, dozens of operators in all, but this is the only one we told participants about.)

The basic retrieval interface was extended with two additional windows: an "aspect window" to help the user collect and annotate found aspects, and (for AspInquery Plus) a 3-D visualization of document relationships.

9.1.1 Aspect Window

With a basic IR system, an analyst may be able to find the documents containing various aspects, but he or she has to use another window or a piece of paper to keep track of what has been found already. We implemented an "aspect window" tool to help with this task. The idea is to provide an area where documents on a particular aspect can be stored. To help label the information, statistical analysis of word and phrase occurrences is used to decide what terms and phrases are most distinctive about a document or set of documents in an aspect. We provided an area for the user to manually assign additional keywords or labels if needed.

Each area of the aspect window has a colored border, a text field at the top for entering a descriptive label, and an automatically generated list of the five noun phrases that most distinguish the group of documents assigned to this aspect from the remainder of the collection. The description field is solely for the user's convenience and need not be filled. If the user wants a description they can type or paste into it, or drag automatically generated phrases into it. The top of Figure 5 shows an example of the aspect window.

9.1.2 Visualization: 3-D Window

Another important step in the aspect oriented retrieval task is deciding (repeatedly) which document to look at next. Aspects represent different forms of relevance, and we believe that they will group together within the set of retrieved documents. AspInquery Plus compares retrieved documents in an extremely high-dimensional space (approximately 400,000 for this collection) where each dimension corresponds to a feature in the collection and the distance was measured by the sine of the angle between the vectors. That space was collapsed to three dimensions for visualization using a spring embedding algorithm.

In the 3-D visualization of the retrieved set, documents that have not been assigned to any aspect have the same blue/purple (read/unread) color scheme that is used in the main window. Documents in the 3-D window are persistent between queries: when new documents are retrieved they are colored light blue (light purple when read) and are placed in the 3-D window by the forces exerted from already placed documents. The bottom of Figure 5 shows five newly retrieved documents in light gray. It is easy to see that three of these documents fall into a group of two previously seen documents (upper right of figure) and the other

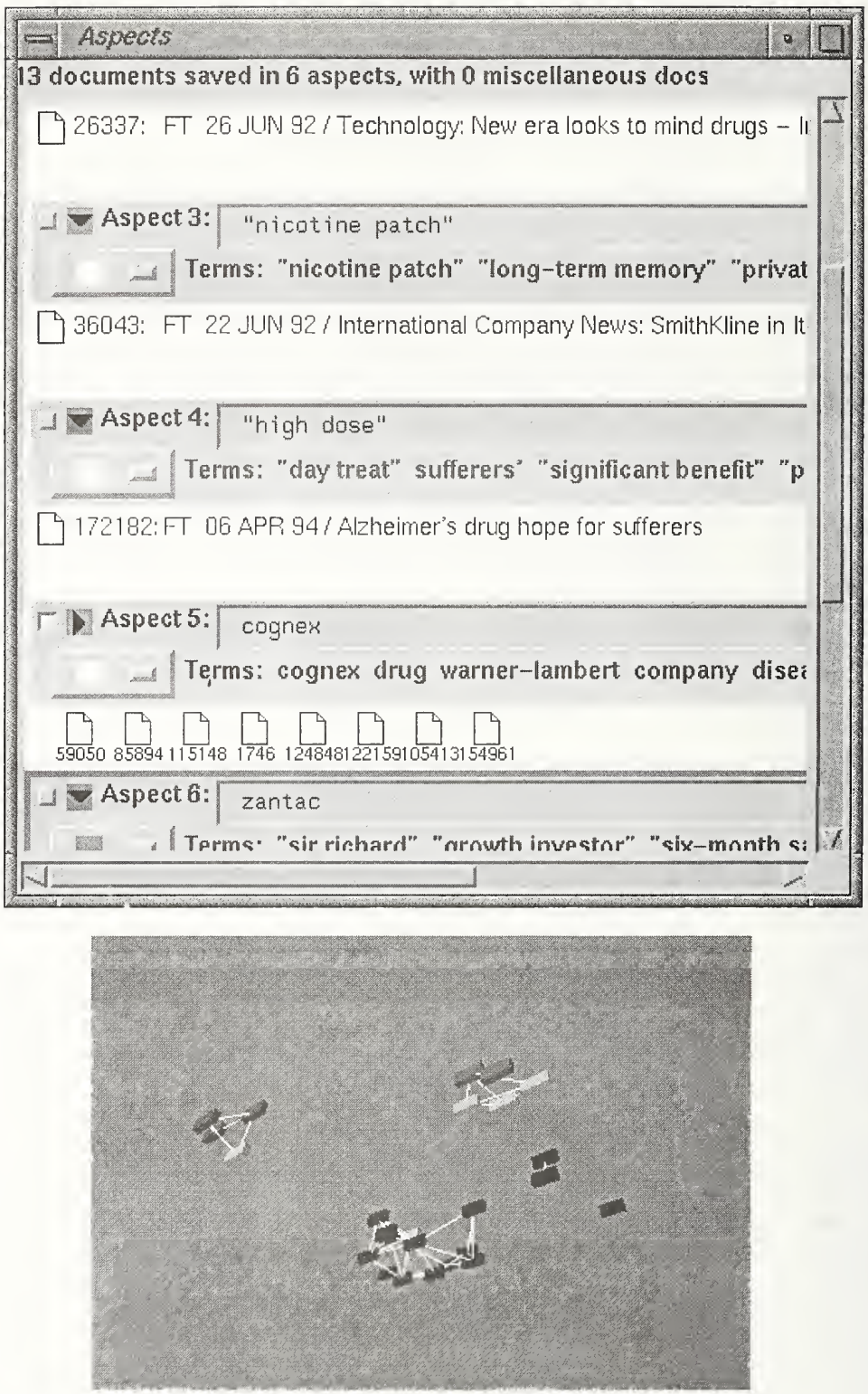


Figure 5: Visualizations provided by the interactive system. Aspect window for interactive system. The top box is the aspect window; the lower figure is the 3-D display

Group	Type	Control	Experimental	Size
1	General	ZP	AI	4
2	Librarian	ZP	AI	4
3	General	ZP	AI+	4
4	Librarian	ZP	AI+	4
5	General	AI	AI+	4

Table 8: Breakdown of participants by systems used for the interactive track

new documents fall into the small group in the upper left and the large group. An analyst who is under time pressure could use the 3-D display to decide that the unjudged document near that aspect is probably on the same aspect and so not worth examining. A retrieved document that is far from any already-marked aspect is more likely to be useful.

9.2 Participants for interactive task

We were interested in how librarians perform search tasks as compared to a more general user population. To that end, we recruited 20 participants: eight librarians and 12 general users. Table 8 shows the types of participants in and the systems used by the different groups in the experiment. Participants were told that the study would take about 3-1/2 hours and that they would be paid $35 if they completed it.

Seven of eight librarians were over 40; six of eight were women; all has very substantial experience with online searching, though had little experience with ranked lists or relevance feedback. The general participants were with one exception under 40; five of the twelve were women; they had moderate to no experience with on-line searching.

9.3 Interactive procedure

The experiment was run in the CIIR's usability laboratory. A "facilitator" was in the room with the participant all of the time except while the participant was doing the tutorials. The same person acted as facilitator for all participants except for the last two in group 5.

First, each participant filled out a questionnaire to give us basic demographic information (age, gender, degrees, general computer experience, experience with various types of searching, etc.). Each participant also took two standard psychometric tests from ETS: a test of verbal fluency (Controlled Associations, test FA-1), and a test of structural visualization (Paper Folding, test VZ-2).

Next, the participant was given a tutorial to learn one system, then they worked on the first three topics. After a short break they were given a tutorial on another system, then they worked on the other three topics. Each search had a 20-minute time limit, and the participant was instructed to stop the search if they had not finished in 20 minutes.

We gave each participant a piece of scratch paper before each search, and a short questionnaire after each. After all the searches were finished the participant was given a final questionnaire, and then "debriefed". The study was conducted single blind: the participants were not told until the debriefing which system was the control and which was the experimental system.

We ran each participant through the entire study in a single essentially continuous period of slightly over three to slightly over four hours, with no breaks longer than about 15 minutes.

9.4 Interactive results

The results are portrayed in Figure 6, a pair of graphs generated and provided by NIST.

9.5 Interactive analysis

Figure 7 shows the amount of variance that can be attributed to: topic, site, system, searcher, and random effects. This is based on a preliminary analysis of the data supplied by NIST of the 52 participants who

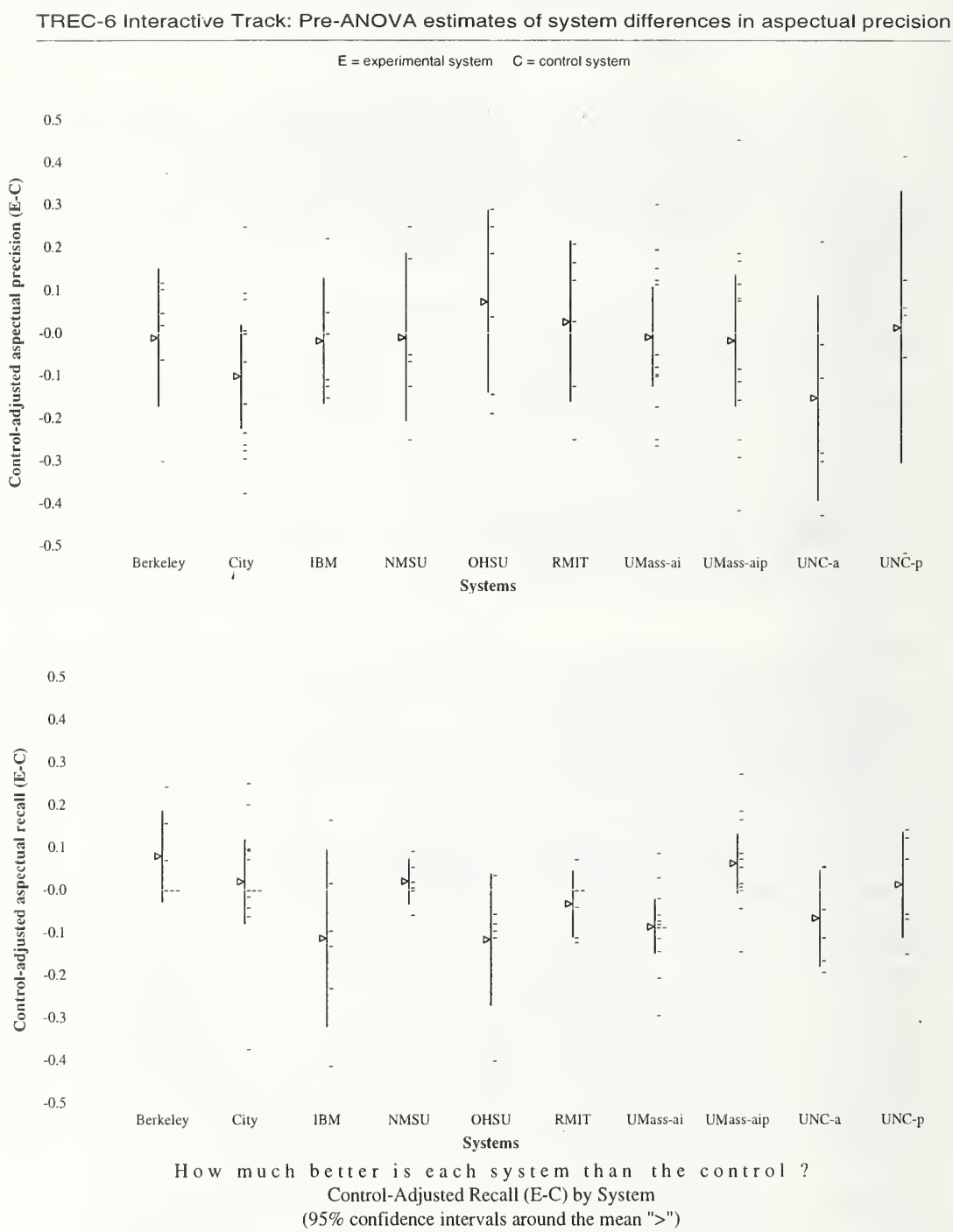


Figure 6: Graphic presentations of pre-ANOVA estimates of system differences via the control. Top graph describes precision; bottom graph, recall. (Provided by NIST.)

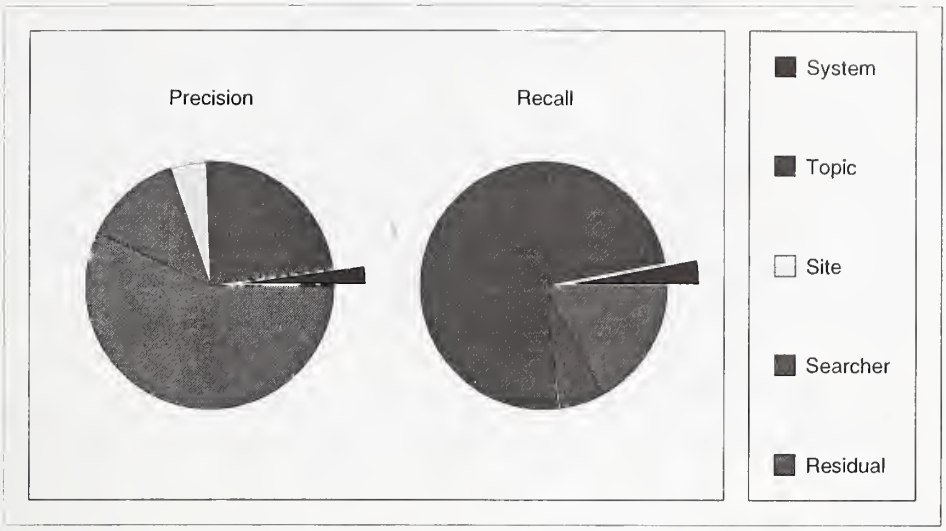


Figure 7: Sources of variance in interactive track, across all sites and all systems.

used ZPRISE as a control. The system differences are small relative to other sources of variation. Statistical analysis (ANOVA) has been performed by both NIST and CIIR, but whether or not statistically significant differences between systems was found depends on which test was used. In the following discussion "significance" claims are based on the tests showing significant differences between systems. Whether or not these differences really exist is discussed in the next subsection.

Figure 6 shows that most systems did not perform significantly different from the control. But at the CIIR *both* of our systems performed significantly different than the control, one worse and one better. (Part of the difference is we have a smaller confidence interval, as we ran 8 users per system and most sites ran 4 users per system.)

For the interactive task, the precision and recall scores are based on the relevance of documents that the searchers marked as being relevant. As a result, precision should be expected to be high. Precision would only be less than 1.0 if the searcher misunderstood a specific topic or made an error. The system effects should be small. Even if a system retrieved a very low precision set, the user must decide which documents are relevant. As can be seen from Figure 6, no system had a significant difference from ZPrise in precision.

For a set to have high aspectual recall, the system must retrieve documents representing all or most of the relevant aspects. The user must then judge those documents, and then save them. The recall score is then based on the recall of retrieved documents, the recall of the documents that are viewed, and the recall of the documents that are saved.

Our 2 systems differed from ZPRISE primarily in the interface presented to the user after a query was run. (The users were instructed in the use of the phrase operator, but most did not use it. Only 6 of the 16 participants in the main groups used it at all, and it was used on only 9 topics.) The aspect window had no features which would be expected to enhance recall. We expected no difference in recall between the AI system and ZPrise. As seen in the bottom graph in Figure 6, AI showed a significant drop in recall versus ZPrise. We are unsure of the reason for this drop. A possible explanation is that the interface is more complicated than the interface for ZPrise, and users had time trouble. We do not believe that this accounts for the difference.

Figure 8 shows the recall of the AI vs ZP for the 2 separate groups. The general group preferred AI over ZP 3 to 1, yet they did significantly worse with AI. The group of librarians preferred ZP over AI 4 to 0. They also did better with ZPrise than with AI, but by a smaller margin (AI outperforming ZPrise is within the confidence interval). Figure 9 shows the difference in time between the experimental system and the control system for the different groups. Group 1, which did significantly worse with AI, had no difference in time required between the two systems so time was clearly not a factor. Group 2 (librarians) did take

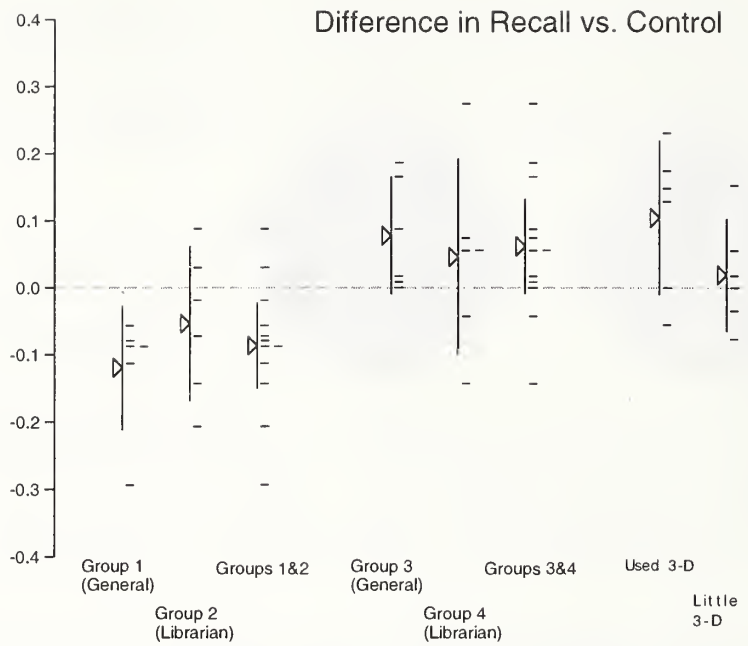


Figure 8: Recall broken down by system compared to control, and by groups of users within system.

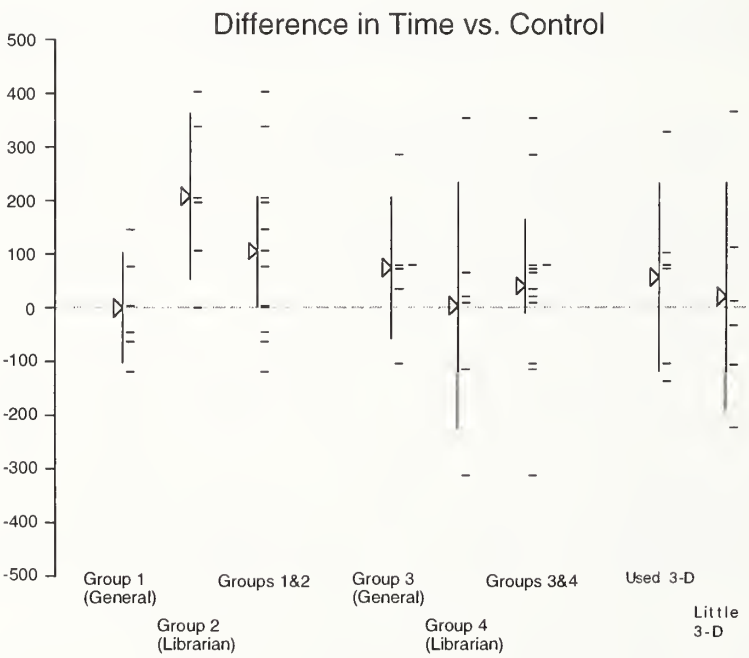


Figure 9: Difference in time spent on task broken down by system compared to control, and by groups of users within system.

Group	Number	Interactions
General	11	4
	12	181
	13	52
	15	0
Librarian	10	0
	14	10
	17	44
	19	143
Final group	16	2
	18	373
	20	148
	21	267

Table 9: Number of interactions with the 3-D window for 12 different users of the system.

signifantly longer with AI than with ZP (200 seconds on average). Time pressure may have been a factor with this group. However, this group did better on recall than group 1.

The other visualization used in our system, the 3-D window, was intended as a recall enhancing device. After a small number of documents have been viewed the 3-D map can be used to select documents that are likely to present new information, and can give better clues than a ranked list. This system showed a significant increase in recall versus ZPrise, and a very large increase in recall compared to AI. We expected an increase in recall with this interface, but we were surprised by the magnitude of the increase. We had learned from previous experience that users are often uncomfortable with 3-D interfaces and may not use them. We instrumented the 3-D window to record user interactions. Table 9 shows the number of interactions with the 3-D window for the different users. If the user ignored the window completely, the system he or she was using was the basic AI system with additional screen clutter. We would expect the results for users who did not use the 3-D window to be consistent with performance on the AI system. We divided the 8 participants into two groups, those who used the 3-D significantly (12, 13, 14, and 19), and those who didn't (11, 15, 10 and 17). (Participant 17 used the 3-D more than participant 14, but that breakdown did not complete the latin square design). The right-hard two bars of Figure 8 show the results for these groups. The group that used the 3-D had higher recall, and the group that didn't use 3-D had similar recall between AI+ and ZP.

9.6 Interactive methodology

NIST performed ANOVA results are reported elsewhere.NIST performed ANOVA on the averaged differences between the experimental systems and the control system (E-C) within each 2x2 Latin Square. The results show a significant difference between experimental systems across all sites, with p = .0133. Pairwise comparisons between systems were done using Tukey's Studentized Range. At p = 0.10, no significant differences were found pairwise between systems. The difference between the AI and AIP was 0.14825. For statistical significance at the 0.10 level, a difference between systems of 0.15033 was required. The obtained difference was 98.6

When the same analysis was performed on just the UMass data that compared a system against ZPRISE, the difference between the E-C data was .14825, for an F-value of 10.99, significant at p = 0.0035. When ANOVA was run on the model

y(i,j,k) = m + s(i) + t(j) + p(k) + e(i,j,k)

with y(i,j,k) being the recall value for searcher k using system i on topic j, we obtained an F value of 3.90, significant at p = 0.0245. The contrast between experimental systems AI and AIP showed a sum of squares of 0.13187 out of a total system based sum of squares of 0.13564.

The ANOVA performed by NIST showed the two UMass systems barely missing significance. The same analysis performed on just UMass data, and a different ANOVA on UMass data both show significance.

One of the hopes of the interactive track is that comparing systems against a common control will provide

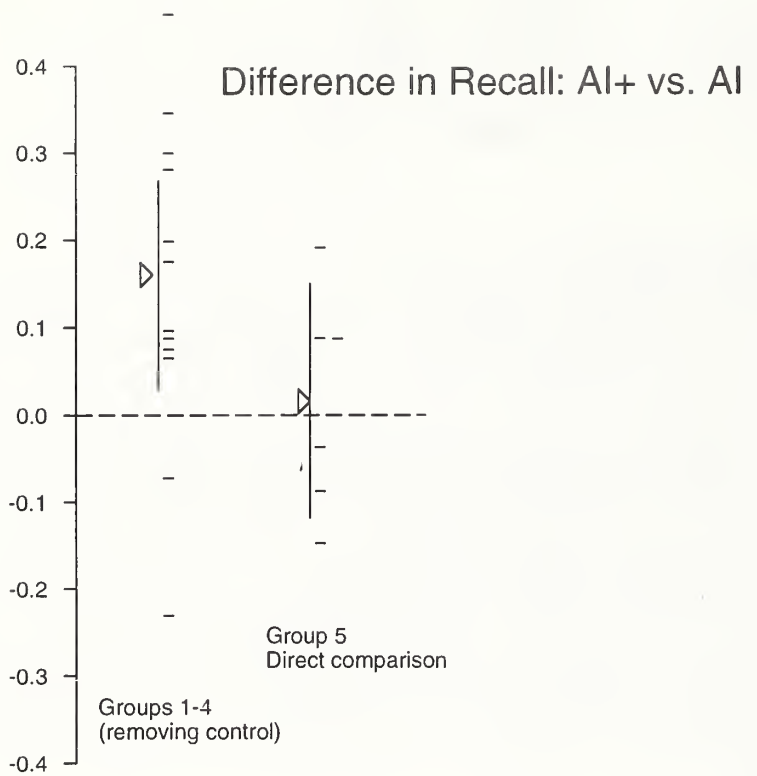


Figure 10: Difference in recall using AspInquery Plus as compared to AspInquery.

the same information as comparing two systems directly against each other. The pre-experiment was designed to validate this approach, but the results are inconclusive. We ran our two experimental systems directly against each other. From the results seen with ZPRISE as a control, we would predict that a significant difference in recall would be observed between the two systems. We did not obtain this result. We found that AI+ outperformed AI in recall with an average value of 0.0156, instead of the 0.14825 value given by the earlier experiment. These results are shown in Figure 10. ANOVA on the direct comparison showed an F value of 0.09, which is not significant. These results, combined with the inconclusive results in the preexperiment, raise questions about the validity of the approach taken in the interactive track.

ANOVA of the results on all five groups of participants showed a difference between systems with an F value of 2.49, $p = 0.089$.

9.7 Interactive conclusions

The system effects were observed with both librarians and a general population. The effects were attenuated on librarians.

Our two systems were much more like each other than they were like the control, but we obtained opposite effects. Since the only difference between the two systems was the 3-D window, we can conclude that providing a graphical display of document similarities as an alternative interface to a ranked list enhances recall in an interactive setting.

Analysis of Variance was performed on our data in several ways, and we obtained varying results. It appears that there is a (marginally) significant difference between our systems, but it is only apparent when measured against a control and is not apparent in direct comparisons. This raises questions about the assumptions and methodology used in the interactive track.

Acknowledgements

We thank Margie Connell, Aiqun Du, Victor Lavrenko, Anton Leouski, Daniella Malin, Darren Mas, Michael Scudder, and Kamal Souccar of the CIIR, as well as Steven Wegmann of Dragon Systems for their assistance in the work described here.

This study is based on research support by several grants: the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623; the National Science Foundation under grant number IRI-9619117; and the NSF Center for Intelligent Information Retrieval at the University of Massachusetts, Amherst.

Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

[1] J. Allan. Incremental relevance feedback. In *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval*, pages 270–278, Zurich, 1996. Association for Computing Machinery.

[2] J. Allan, J. P. Callan, W. B. Croft, L. Ballesteros, J. Broglio, J. Xu, and H. Shu. INQUERY at TREC-5. In D. Harman, editor, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*. National Institute of Standards and Technology Special Publication, (in press).

[3] Lisa Ballesteros and W. Bruce Croft. Dictionary-based methods for cross-lingual information retrieval. In *Proceedings of the 7th International DEXA Conference on Database and Expert Systems Applications*, pages 791–801, 1996.

[4] Lisa Ballesteros and W. Bruce Croft. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 84–91, Philadelphia, 1997. Association for Computing Machinery.

[5] E. W. Brown. Fast evaluation of structured queries for information retrieval. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval*, pages 30–38, Seattle, 1995. Association for Computing Machinery.

[6] Chris Buckley and Gerard Salton. Optimization of relevance feedback weights. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval*, pages 351–357, Seattle, Washington, July 1995. ACM.

[7] Don Byrd, Russell Swan, and James Allan. TREC-6 interactive track report, part 1: Experimental procedure and initial results. Technical Report IR-117, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, November 1997.

[8] J. P. Callan. Document filtering with inference networks. In *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval*, pages 262–269, Zurich, 1996. Association for Computing Machinery.

[9] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, pages 78–83, Valencia, Spain, 1992. Springer-Verlag.

[10] Mark W. Davis and William C. Ogden. Quilt: Implementing a large-scale cross-language text retrieval system. In *Proceedings of the 20th International Conference on Research and Development in Information Retrieval*, pages 92–98, 1997.

[11] Warren R. Greiff, W. Bruce Croft, and Howard Turtle. Computationally tractable probabilistic modeling of boolean operators. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 119–128, Philadelphia, 1997. Association for Computing Machinery.

[12] David A. Hull and Gregory Grefenstette. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pages 49–57, 1996.

[13] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval*, pages 4–11, Zurich, 1996. Association for Computing Machinery.

A CLIR Track Questionnaire

A.1 OVERALL APPROACH:

A.1.1 What basic approach do you take to cross-language retrieval?

[x] Query Translation
[ ] Document Translation
[ ] Other, ----------

A.1.2 Were manual translations of the original NIST topics used as a starting point for any of your cross-language runs?

[ ] No
[x] Yes, translated by a native spanish speaker then submitted to trec

A.1.3 Were the automatically translated (Logos MT) documents used for any of your cross-language runs?

[x] No
[ ] Yes, ----------

A.1.4 Were the automatically translated (Logos MT) topics used for any of your cross-language runs?

[x] No
[ ] Yes, ----------

A.2 MANUAL QUERY FORMULATION:

A.2.1 If query formulation involved manual effort, how fluent was the user in the source (query) language?

[ ] ----------

A.2.2 If query formulation involved manual effort, how fluent was the user in the target (document) language?

[ ] ----------

A.3 USE OF MANUALLY GENERATED DATA RESOURCES:

A.3.1 What kind of manually generated data resources were used?

[x] Dictionaries
[ ] Thesauri
[ ] Part-of-speech Lists
[x] Other, UN aligned corpus

A.3.2 Were they generated with information retrieval in mind or were they taken from related fields?

[ ] Information Retrieval
[ ] Machine Translation
[ ] Linguistic Research
[x] General Purpose Dictionaries
[ ] Other, ----------

A.3.3 Were they specifically tuned for the data being searched (ie. with special terminology) or general-purpose?

[ ] Tuned for data; Please specify _____
[x] General purpose

A.3.4 What amount of work was involved in adapting them for use in your information retrieval system.

[ ] None
[x] involved cleaning mark-up meant for human users

A.3.5 Size

[ ] Collins about 50k_____ entries
[ ] UN data: 500 MBytes

A.3.6 Availability? - Please also provide sources/references!

[ ] Commercial
[x] Proprietary, Collins spanish-english MRD
[ ] Free
[x] Other, UN data from the LDC

A.4 USE OF AUTOMATICALLY GENERATED DATA RESOURCES:

A.4.1 Form of the automatically constructed data resources?

[ ] Lexicon
[x] Thesaurus
[ ] Similarity matrix
[x] Other, phrase dictionary of word usage and phrasal information from dictionary

A.4.2 What sort of training data was used to construct them?

[x] Same data as used for searches, AP database
[x] Similar data as used for searches, El Norte collection
[ ] Other data, _____

A.4.3 Size

[x] Sp. Th: 112k, Phr. Dict:48k entries
[x] Eng. Th: 287, Sp. Th: 74, Phr. Dict: 6 MBytes

A.4.4 Was there any manual clean-up involved in the construction process?

[ ] Yes, _____
[x] No

A.4.5 Rough resource estimates for building the data resources (ie. an indicator of the computational complexity of the process).

[x] 200MB/hour (co-occurrence thesaurus)
[x] 120 MB memory used per 1 Gig. data
[x] about 2x collection size temporary disk space

A.5 GENERAL

A.5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?

[ ] Very dependent, ----------
[ ] Somewhat dependent, ----------
[x] Easily replacable, ----------
[ ] Don't know

A.5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?

[x] Yes, a lot, e.g. specialized dictionaries
[ ] Yes, somewhat, ----------
[ ] No, not significantly, ----------
[ ] Don't know

A.5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?

[ ] Yes a lot, ----------
[x] Yes, somewhat, ----------
[ ] No, not significantly, ----------
[ ] Don't know

A.5.4 Are similar resources available for other languages than those used?

[x] Yes, ----------
[ ] No

B TREC Interactive Track Protocol Log

The following is the log of the interaction for Participant 13, Topic 1 (326i). Spoken words are shown in italics. "U:" (for "User") precedes remarks and actions by the participant; "F:" precedes remarks by the Facilitator. Times are shown as "n s" for n seconds from the start of the session.

Time set at Tue Aug 12 09:24:57 1997
query is ferry sinking casualties
Query is: ferry sinking casualties
bl->term_freq = 0, default_belief = 0.400000, totalhits = 2932
bl->doc_cnt = 20
24 s Tue Aug 12 09:25:21 1997
Number of docs found is 20
1: 177935: FT943-312: FT 30 SEP 94 / Ferries in six 'near accidents': Finland and Sweden order checks after Estonia sinking
2: 205199: FT944-15661: FT 17 OCT 94 / World News in Brief: Bangladesh ferry sinks
3: 174281: FT943-178: FT 30 SEP 94 / Leading Article: Defying the cruel sea
4: 204595: FT944-15057: FT 20 OCT 94 / Improved ferry safety urged
5: 194241: FT944-5773: FT 02 DEC 94 / World News in Brief: Manila ferry sinks
6: 200503: FT944-11367: FT 07 NOV 94 / Pounds 45m car-ferry research planned
7: 199238: FT944-10102: FT 12 NOV 94 / Tighter ferry rules proposed
8: 208111: FT944-18217: FT 05 OCT 94 / World News in Brief: Check on ferries ordered
9: 200184: FT944-11048: FT 08 NOV 94 / Bow doors faulty on 33% of ferries using UK ports: Government to increase safety checks on vessels
10: 208769: FT944-18875: FT 01 OCT 94 / What future for the ferry?: Questions raised by the Baltic tragedy
11: 178166: FT943-543: FT 29 SEP 94 / Bow doors leak reported after 800 die in Baltic ferry sinking
12: 193552: FT944-5084: FT 06 DEC 94 / Ro-ro ferry study agreed
13: 75524: FT931-8485: FT 19 FEB 93 / Crowded ferry sinks off Haiti
14: 199245: FT944-10109: FT 12 NOV 94 / Tighter ferry rules proposed
15: 193716: FT944-5248: FT 05 DEC 94 / Sea safety review focuses on ferries
16: 178159: FT943-536: FT 29 SEP 94 / Safety rules that failed the Estonia: It was a modern ship, well maintained and partly Swedish owned. But are even the best ro-ro ferries vulnerable?
17: 177939: FT943-316: FT 30 SEP 94 / Ferries face calls for safety curbs: Estonia disaster brings reports of other 'near accidents'
18: 199989: FT944-10853: FT 09 NOV 94 / Eurotunnel hits at government on ferry safety
19: 39232: FT923-4546: FT 05 SEP 92 / Swan wins order for Tyne ferry
20: 39203: FT923-4517: FT 05 SEP 92 / Ferry order for Tyne yard
38 s, Reading doc 177935:FT943-312, click from main win, time Tue Aug 12 09:25:35 1997
0:42 U: *OK, so my first article is about a ferry that sank and 900 people died.*
71 s, Reading doc 177935:FT943-312, click from main win, time Tue Aug 12 09:26:08 1997
Doc number 177935 added to aspect 0
Aspect # 0, auto terms are estonia "estonia sink" forsberg "estonia disaster" "bow section"
No user supplied text
1:18 U:*This article describes six incidents.*
1:30 F: *Six incidents of ferry sinkings?*
U: *Right.*
1:48 U: *Oh, talks about six near accidents and it describes one that actually happened.*
148 s, Reading doc 205199:FT944-15661, click from main win, time Tue Aug 12 09:27:25 1997
Doc number 205199 added to aspect 1
2:30 U: *This is a brief article about 400 people that dies in Bangladesh so we'll save that, and the name ... we'll name that one Bangladesh.*
Aspect # 1, auto terms are "ferry sink" "ferry disaster" "wedding party" "high sea" bangladesh
U: drags "bangladesh" into label area
197 s, Reading doc 174281:FT943-178, click from main win, time Tue Aug 12 09:28:14 1997
3:18 U: *Here's another article about the Estonia incident. It's a repeat so I don't need to save that, or should I save that also under*
F: *No, there's no need, there's no point to saving additional ones.*

224 s, Reading doc 204595:FT944-15057, click from main win, time Tue Aug 12 09:28:41 1997
3:45 U: *This one just talks about the Estonia again, so we don't need that.*
232 s, Reading doc 194241:FT944-5773, click from main win, time Tue Aug 12 09:28:49 1997
3:53 U: *And the fifth one is in Manila, 480 people were on it, 275 were rescued, and they're still picking up survivors, so you can probably assume 100 people died, so go ahead and save it.*
Doc number 194241 added to aspect 2
Aspect # 2, auto terms are "ferry sink" "cargo ship" manila sink survivor
U: drags "manila" into label area
267 s, Reading doc 200503:FT944-11367, click from main win, time Tue Aug 12 09:29:24 1997
272 s, Reading doc 199238:FT944-10102, click from main win, time Tue Aug 12 09:29:29 1997
4:34 U: *A lot of these keep talking about tighter regulations due to the sinking of the Estonia.*
288 s, Reading doc 200184:FT944-11048, click from main win, time Tue Aug 12 09:29:45 1997
Doc number 200184 added to aspect 3
Aspect # 3, auto terms are "bow door" "marine safety agent" ferry "safety agent" "dr mawhinney"
4:57 U: *Here's one that briefly mentions a ship...*
5:09 U: *This one again is more about safety regulations, but it briefly mentions a ship that had 193 casualties, so I guess I'll type in my own word since the one I want isn't in there.*
U: types " Herald of Free Enterprise" in label area.
5:38 U: *I'm trying to name all these either by the name of the ship or where it happened.*
U: moves controls on 3-D window and alters view several times.
5:54 U: *I'm trying to see if I can use the 3-D to help me out*
F: *I'm sorry, you couldn't couldn't what... You're trying to see if*
U: *I'm trying to see if I can use this to give me ... I'm assuming that these are supposed to show articles in the connecting blocks that are more relevant*
F: *That are more similar to each other*
U: *More similar.*
407 s, Reading doc 208769:FT944-18875, click from main win, time Tue Aug 12 09:31:44 1997
421 s, Reading doc 178166:FT943-543, click from main win, time Tue Aug 12 09:31:59 1997
448 s, Reading doc 193552:FT944-5084, click from main win, time Tue Aug 12 09:32:25 1997
454 s, Reading doc 75524:FT931-8485, click from main win, time Tue Aug 12 09:32:31 1997
Doc number 75524 added to aspect 4
Aspect # 4, auto terms are "ferry sink" port-au-prince neptune haiti "product centre"
U: drags "neptune" into label area
485 s, Reading doc 199245:FT944-10109, click from main win, time Tue Aug 12 09:33:02 1997
491 s, Reading doc 193716:FT944-5248, click from main win, time Tue Aug 12 09:33:08 1997
493 s, Reading doc 177939:FT943-316, click from main win, time Tue Aug 12 09:33:10 1997
500 s, Reading doc 199989:FT944-10853, click from main win, time Tue Aug 12 09:33:17 1997
505 s, Reading doc 39232:FT923-4546, click from main win, time Tue Aug 12 09:33:22 1997
512 s, Reading doc 39203:FT923-4517, click from main win, time Tue Aug 12 09:33:29 1997
8:32 U: *So I went through all 20 of the articles. For the most part I'd say all but probably 3 or 4 talked about accidents with over 100 casualties, so should I try a new search?*
F: *It's up to you. You have plenty of time.*
U: *You mean try a different wording of it?*
F: *It's up to you.*
9:20 F: *You could also try raising the max docs.*
U: *OK.*
query is ferry sinking casualties
Query is: ferry sinking casualties
bl->term_freq = 0, default_belief = 0.400000, totalhits = 2932
bl->doc_cnt = 40
574 s Tue Aug 12 09:34:31 1997
Number of docs found is 40
1: 177935: FT943-312: FT 30 SEP 94 / Ferries in six 'near accidents': Finland and Sweden order checks after Estonia sinking

2: 205199: FT944-15661: FT 17 OCT 94 / World News in Brief: Bangladesh ferry sinks
3: 174281: FT943-178: FT 30 SEP 94 / Leading Article: Defying the cruel sea
4: 204595: FT944-15057: FT 20 OCT 94 / Improved ferry safety urged
5: 194241: FT944-5773: FT 02 DEC 94 / World News in Brief: Manila ferry sinks
6: 200503: FT944-11367: FT 07 NOV 94 / Pounds 45m car-ferry research planned
7: 199238: FT944-10102: FT 12 NOV 94 / Tighter ferry rules proposed
8: 208111: FT944-18217: FT 05 OCT 94 / World News in Brief: Check on ferries ordered
9: 200184: FT944-11048: FT 08 NOV 94 / Bow doors faulty on 33% of ferries using UK ports: Government to increase safety checks on vessels
10: 208769: FT944-18875: FT 01 OCT 94 / What future for the ferry?: Questions raised by the Baltic tragedy
11: 178166: FT943-543: FT 29 SEP 94 / Bow doors leak reported after 800 die in Baltic ferry sinking
12: 193552: FT944-5084: FT 06 DEC 94 / Ro-ro ferry study agreed
13: 75524: FT931-8485: FT 19 FEB 93 / Crowded ferry sinks off Haiti
14: 199245: FT944-10109: FT 12 NOV 94 / Tighter ferry rules proposed
15: 193716: FT944-5248: FT 05 DEC 94 / Sea safety review focuses on ferries
16: 178159: FT943-536: FT 29 SEP 94 / Safety rules that failed the Estonia: It was a modern ship, well maintained and partly Swedish owned. But are even the best ro-ro ferries vulnerable?
17: 177939: FT943-316: FT 30 SEP 94 / Ferries face calls for safety curbs: Estonia disaster brings reports of other 'near accidents'
18: 199989: FT944-10853: FT 09 NOV 94 / Eurotunnel hits at government on ferry safety
19: 39232: FT923-4546: FT 05 SEP 92 / Swan wins order for Tyne ferry
20: 39203: FT923-4517: FT 05 SEP 92 / Ferry order for Tyne yard
21: 207852: FT944-17958: FT 05 OCT 94 / Finns order ro-ro bow doors welded shut
22: 169325: FT942-14757: FT 19 APR 94 / Letters to the Editor: Channel control overdue
23: 207851: FT944-17957: FT 05 OCT 94 / UN maritime agency panel to review safety: A look at action prompted by the Baltic ferry disaster
24: 208393: FT944-18499: FT 04 OCT 94 / Baltic ferry operators to weld bow doors shut: Safety move follows confirmation of cause of Estonia disaster
25: 208098: FT944-18204: FT 05 OCT 94 / Maritime agency in safety plan
26: 208402: FT944-18508: FT 04 OCT 94 / Estonia's bow doors were torn off in heavy storm: Video of sunken ferry shows how water flooded car deck
27: 204681: FT944-15143: FT 19 OCT 94 / Estonia's missing bow door located
28: 200149: FT944-11013: FT 08 NOV 94 / International Company News: Heavy loss in US pushes Trygg-Hansa into the red - Swedish insurer posts SKr813m deficit at nine months
29: 141158: FT941-5434: FT 07 MAR 94 / Freight companies to shun Channel tunnel
30: 208832: FT944-18938: FT 01 OCT 94 / UN agency orders ferry probe: Estonia's bow doors may have been torn off in storm, Swedish authorities say
31: 178158: FT943-535: FT 29 SEP 94 / Tragedy leaves Swedes in shock
32: 171243: FT942-16675: FT 08 APR 94 / Survey of East Kent (7): Pain amid the gain - The ferries fight back
33: 195783: FT944-6974: FT 26 NOV 94 / Thinking the unsinkable: The modern parallels exposed by an exhibition about the Titanic, which sank in 1912
34: 205276: FT944-15738: FT 17 OCT 94 / Company News This Week: Departure delays leave investors counting the cost - Eurotunnel
35: 137406: FT934-1954: FT 16 DEC 93 / Technology: Ships bridge the danger gap - Andrew Fisher concludes a series on transport safety with an investigation into innovations that may help prevent sea disasters and give clues to their causes
36: 127095: FT934-8445: FT 16 NOV 93 / Corporate bankruptcies increase as demand sinks
37: 1655: FT911-4602: FT 18 APR 91 / MMC to investigate Isle of Wight ferries
38: 206611: FT944-1600: FT 19 DEC 94 / Survey of Sweden (14): A remarkable comeback - Profile: Stena Line
39: 26988: FT922-7334: FT 19 MAY 92 / World Trade News: Denmark-Sweden ferry link-up is agreed
40: 119826: FT933-1606: FT 23 SEP 93 / Ferry operator in link with Belgium
U: Several 3-D interactions
Reading doc 200503: FT944-11367, click from 3-D window
642 s, Reading doc 207852:FT944-17958, click from main win, time Tue Aug 12 09:35:39 1997
652 s, Reading doc 169325:FT942-14757, click from main win, time Tue Aug 12 09:35:49 1997
661 s, Reading doc 207851:FT944-17957, click from main win, time Tue Aug 12 09:35:58 1997
11:11 U: *So I increased the maxdocs from 20 to 40, and most of the later articles don't seem to really have much relevant information. Either they're talking*

about the Estonia or they're just talking about general safety regulations.
678 s, Reading doc 208393:FT944-18499, click from main win, time Tue Aug 12 09:36:15 1997
701 s, Reading doc 208098:FT944-18204, click from main win, time Tue Aug 12 09:36:38 1997
11:45 U: *I'm guessing that's why there's this big network here.* (Points to large cluster of documents in 3-D viewer.) *A lot of them are talking about the Estonia so I think they're all related in that sense.*
723 s, Reading doc 208402:FT944-18508, click from main win, time Tue Aug 12 09:37:00 1997
734 s, Reading doc 204681:FT944-15143, click from main win, time Tue Aug 12 09:37:11 1997
737 s, Reading doc 200149:FT944-11013, click from main win, time Tue Aug 12 09:37:14 1997
744 s, Reading doc 141158:FT941-5434, click from main win, time Tue Aug 12 09:37:21 1997
12:32 U: *Yeah this is really starting to get ... My query is "ferry sinking", and in this article the word "sink" only appears once, and it doesn't have anything to do with ferries, and there's nothing about casualties so it looks like we're getting farther and farther away from anything relevant. You can see that over here, we're moving further away from this point.* (Points to several documents in 3-D view)
775 s, Reading doc 208832:FT944-18938, click from main win, time Tue Aug 12 09:37:52 1997
800 s, Reading doc 171243:FT942-16675, click from main win, time Tue Aug 12 09:38:17 1997
813 s, Reading doc 195783:FT944-6974, click from main win, time Tue Aug 12 09:38:30 1997
822 s, Reading doc 205276:FT944-15738, click from main win, time Tue Aug 12 09:38:39 1997
827 s, Reading doc 137406:FT934-1954, click from main win, time Tue Aug 12 09:38:44 1997
13:49 U: *OK, I've found a new one.*
Doc number 137406 added to aspect 5
Aspect # 5, auto terms are moby imo vessel livorno ship
U: drags "livorno" into label area
14:06 U: *This is the first new article I've found in the last 20 I've looked at.*
868 s, Reading doc 127095:FT934-8445, click from main win, time Tue Aug 12 09:39:25 1997
872 s, Reading doc 1655:FT911-4602, click from main win, time Tue Aug 12 09:39:29 1997
876 s, Reading doc 206611:FT944-1600, click from main win, time Tue Aug 12 09:39:33 1997
884 s, Reading doc 26988:FT922-7334, click from main win, time Tue Aug 12 09:39:41 1997
892 s, Reading doc 119826:FT933-1606, click from main win, time Tue Aug 12 09:39:49 1997
15:00 F: *You have five minutes.*
15:23 U: *We'll try searching for ferry and accidents.*
query is ferry accident
Query is: ferry accident
bl->term_freq = 0, default_belief = 0.400000, totalhits = 1978
bl->doc_cnt = 40
932 s Tue Aug 12 09:40:29 1997
Number of docs found is 40
1: 174533: FT943-3295: FT 15 SEP 94 / Inquiry starts after six die in ferry walkway collapse
2: 42744: FT923-7671: FT 15 AUG 92 / Deaths ferry to be withdrawn
3: 149044: FT941-12581: FT 29 JAN 94 / Accident halts ferry services
4: 72637: FT931-5947: FT 03 MAR 93 / World News in Brief: Congo ferry toll rises to 146
5: 177935: FT943-312: FT 30 SEP 94 / Ferries in six 'near accidents': Finland and Sweden order checks after Estonia sinking
6: 186187: FT943-1246: FT 26 SEP 94 / World News in Brief: 16 injured in lifeboat accident
7: 208393: FT944-18499: FT 04 OCT 94 / Baltic ferry operators to weld bow doors shut: Safety move follows confirmation of cause of Estonia disaster
8: 208402: FT944-18508: FT 04 OCT 94 / Estonia's bow doors were torn off in heavy storm: Video of sunken ferry shows how water flooded car deck
9: 9804: FT921-686: FT 27 MAR 92 / Crash probe finds 'no abnormality'
10: 174478: FT943-3240: FT 15 SEP 94 / Investigators widen probe on ferry walkway collapse
11: 186180: FT943-1239: FT 26 SEP 94 / World News in Brief: 16 injured in lifeboat accident
12: 207852: FT944-17958: FT 05 OCT 94 / Finns order ro-ro bow doors welded shut
13: 177939: FT943-316: FT 30 SEP 94 / Ferries face calls for safety curbs: Estonia disaster brings reports of other 'near accidents'
14: 201958: FT944-12822: FT 31 OCT 94 / Business Travel: In S Korea, it is better to arrive ..
15: 208769: FT944-18875: FT 01 OCT 94 / What future for the ferry?: Questions raised by the Baltic tragedy
16: 14222: FT921-11074: FT 03 FEB 92 / UK Company News: Eurotunnel to seek damages for cost of extra safety

17: 178552: FT943-6917: FT 26 AUG 94 / Cross-Channel ferry blaze to be investigated
18: 1655: FT911-4602: FT 18 APR 91 / MMC to investigate Isle of Wight ferries
19: 5733: FT921-365: FT 30 MAR 92 / Hopes for ship data recorder
20: 26988: FT922-7334: FT 19 MAY 92 / World Trade News: Denmark-Sweden ferry link-up is agreed
21: 119826: FT933-1606: FT 23 SEP 93 / Ferry operator in link with Belgium
22: 150782: FT941-1125: FT 26 MAR 94 / International Company News: Vard plans to spin off ferry division
23: 118260: FT933-15867: FT 07 JUL 93 / New high-speed Stena ferry in service by 1995
24: 119845: FT933-1625: FT 23 SEP 93 / Sally Line agrees Belgian link-up
25: 32757: FT922-12800: FT 15 APR 92 / Freight ferry
26: 199245: FT944-10109: FT 12 NOV 94 / Tighter ferry rules proposed
27: 114042: FT933-11894: FT 27 JUL 93 / International Company News: Vard set to spin off ferry unit
28: 200184: FT944-11048: FT 08 NOV 94 / Bow doors faulty on 33% of ferries using UK ports: Government to increase safety checks on vessels
29: 62351: FT924-11264: FT 27 OCT 92 / Ferry operators accused of pricing collusion
30: 199989: FT944-10853: FT 09 NOV 94 / Eurotunnel hits at government on ferry safety
31: 143053: FT941-732: FT 29 MAR 94 / Netherlands ferry route may restart
32: 199238: FT944-10102: FT 12 NOV 94 / Tighter ferry rules proposed
33: 84325: FT931-16573: FT 06 JAN 93 / Cross-Channel ferries hint
34: 125074: FT934-497: FT 24 DEC 93 / International Company News: Greek ferry operator in cash call
35: 64622: FT924-13535: FT 15 OCT 92 / New ferry service
36: 28865: FT922-9211: FT 08 MAY 92 / New ferry is largest in Channel
37: 137406: FT934-1954: FT 16 DEC 93 / Technology: Ships bridge the danger gap - Andrew Fisher concludes a series on transport safety with an investigation into innovations that may help prevent sea disasters and give clues to their causes
38: 2421: FT911-5368: FT 15 APR 91 / World News in Brief: Ferries disrupted
39: 26728: FT922-7074: FT 20 MAY 92 / Boulogne freight link
40: 44107: FT923-9034: FT 07 AUG 92 / Ferry row settled
951 s, Reading doc 174533:FT943-3295, click from main win, time Tue Aug 12 09:40:48 1997
956 s, Reading doc 42744:FT923-7671, click from main win, time Tue Aug 12 09:40:53 1997
965 s, Reading doc 149044:FT941-12581, click from main win, time Tue Aug 12 09:41:02 1997
970 s, Reading doc 72637:FT931-5947, click from main win, time Tue Aug 12 09:41:07 1997
Doc number 72637 added to aspect 6
Aspect # 6, auto terms are congo brazzaville zairean "illegal immigrant" "death toll"
U: drags "congo" into label area
992 s, Reading doc 208393:FT944-18499, click from main win, time Tue Aug 12 09:41:29 1997
1006 s, Reading doc 208402:FT944-18508, click from main win, time Tue Aug 12 09:41:43 1997
1008 s, Reading doc 9804:FT921-686, click from main win, time Tue Aug 12 09:41:45 1997
1012 s, Reading doc 186180:FT943-1239, click from main win, time Tue Aug 12 09:41:49 1997
1016 s, Reading doc 207852:FT944-17958, click from main win, time Tue Aug 12 09:41:53 1997
1020 s, Reading doc 177939:FT943-316, click from main win, time Tue Aug 12 09:41:57 1997
1022 s, Reading doc 201958:FT944-12822, click from main win, time Tue Aug 12 09:41:59 1997
1031 s, Reading doc 208769:FT944-18875, click from main win, time Tue Aug 12 09:42:08 1997
1034 s, Reading doc 14222:FT921-11074, click from main win, time Tue Aug 12 09:42:11 1997
1041 s, Reading doc 178552:FT943-6917, click from main win, time Tue Aug 12 09:42:18 1997
1048 s, Reading doc 1655:FT911-4602, click from main win, time Tue Aug 12 09:42:25 1997
1054 s, Reading doc 5733:FT921-365, click from main win, time Tue Aug 12 09:42:31 1997
1064 s, Reading doc 26988:FT922-7334, click from main win, time Tue Aug 12 09:42:41 1997
1070 s, Reading doc 119826:FT933-1606, click from main win, time Tue Aug 12 09:42:47 1997
1079 s, Reading doc 118260:FT933-15867, click from main win, time Tue Aug 12 09:42:56 1997
1082 s, Reading doc 119845:FT933-1625, click from main win, time Tue Aug 12 09:42:59 1997
1089 s, Reading doc 32757:FT922-12800, click from main win, time Tue Aug 12 09:43:06 1997
1092 s, Reading doc 199245:FT944-10109, click from main win, time Tue Aug 12 09:43:09 1997
1096 s, Reading doc 114042:FT933-11894, click from main win, time Tue Aug 12 09:43:13 1997
1105 s, Reading doc 125074:FT934-497, click from main win, time Tue Aug 12 09:43:22 1997
U: does extensive interactions with 3-D window
19:06 F: *Could you say what you're doing there? With the 3-D window?*
U: *I'm just looking at it and trying to see how the articles I've picked lay out in this 3-D network. I'm just trying to figure out how I could make it more*

useful for my searching purposes. I'm really thinking about things how if I'm searching for things on the Internet and I had something like this how would I be able to use it. It's an interesting idea.
20:00 F: *Time's up.*
7 documents saved in 7 aspects, with 0 miscellaneous docs
Aspect 0, 1 docs saved
Auto Terms: estonia estonia_sink forsberg estonia_disaster bow_section
User supplied text = estonia
FT943-312: 177935 FT 30 SEP 94 / Ferries in six 'near accidents': Finland and Sweden order checks after Estonia sinking
Aspect 1, 1 docs saved
Auto Terms: ferry_sink ferry_disaster wedding_party high_sea bangladesh
User supplied text = bangladesh
FT944-15661: 205199 FT 17 OCT 94 / World News in Brief: Bangladesh ferry sinks
Aspect 2, 1 docs saved
Auto Terms: ferry_sink cargo_ship manila sink survivor
User supplied text = manila
FT944-5773: 194241 FT 02 DEC 94 / World News in Brief: Manila ferry sinks
Aspect 3, 1 docs saved
Auto Terms: bow_door marine_safety_agent ferry safety_agent dr_mawhinney
User supplied text = Herald of Free Enterprise
FT944-11048: 200184 FT 08 NOV 94 / Bow doors faulty on 33% of ferries using UK ports: Government to increase safety checks on vessels
Aspect 4, 1 docs saved
Auto Terms: ferry_sink port-au-prince neptune haiti product_centre
User supplied text = neptune
FT931-8485: 75524 FT 19 FEB 93 / Crowded ferry sinks off Haiti
Aspect 5, 1 docs saved
Auto Terms: moby imo vessel livorno ship
User supplied text = livorno
FT934-1954: 137406 FT 16 DEC 93 / Technology: Ships bridge the danger gap - Andrew Fisher concludes a series on transport safety with an investigation into innovations that may help prevent sea disasters and give clues to their causes
Aspect 6, 1 docs saved
Auto Terms: congo brazzaville zairean illegal_immigrant death_toll
User supplied text = congo
FT931-5947: 72637 FT 03 MAR 93 / World News in Brief: Congo ferry toll rises to 146
1200 s Tue Aug 12 09:45:21 1997
Stats from this run: 3 queries run
100 docs returned, 66 unique, 52 viewed
7 docs saved (including misc), 7 saved
saved docs:
FT931-5947: 72637 979
FT931-8485: 75524 466
FT934-1954: 137406 843
FT943-312: 177935 75
FT944-5773: 194241 255
FT944-11048: 200184 311
FT944-15661: 205199 162
saved good docs
FT931-5947: 72637 979
FT931-8485: 75524 466
FT934-1954: 137406 843
FT943-312: 177935 75
FT944-5773: 194241 255
FT944-11048: 200184 311
FT944-15661: 205199 162
Sparse Trec Data Starts HERE

1 FT943-312
2 FT944-15661
3 FT944-5773
4 FT944-11048
5 FT931-8485
6 FT934-1954
7 FT931-5947

TREC-6 English and Chinese Retrieval Experiments using PIRCS

K.L. Kwok, L. Grunfeld and J.H. Xu

Computer Science Dept., Queens College, CUNY
Flushing, NY 11367
Email: kwok@ir.cs.qc.edu URL: http://ir.cs.qc.edu

Abstract

For Trec-6 ad-hoc experiments, we continue to use two-stage retrieval with pseudo-feedback from top-ranked un-judged documents for both Chinese and English. We perform three types of retrieval characterized by queries formed using title only, description only and all sections of the given topics. For short queries mainly derived from title or description section, query terms are weighted by average term frequency avtf introduced previously. For Chinese, we employ a combination of representation (character, bigram and short-word) strategy, returning the highest average non-interpolated precision that is even better than some manual approaches. In English ad-hoc, we try a document re-ranking strategy for the first stage retrieval based on occurrence of selected query term pairs, so as to have better result in the second stage. Performance for English ad-hoc is also highly competitive for both very short and long queries.

In routing, a strategy of combining different methods of query formation and retrieval is used. These include no learning ad-hoc type queries, learning from the more current FBIS5 documents only, queries learnt from selecting the best set of known relevant documents based on a genetic algorithm, and queries that are trained from a back-propagation neural network with hidden nodes. Average precision results are among the best four. In addition, we also participate in high precision and the filtering track.

1. Introduction

Results from our PIRCS retrieval engine have been demonstrated to be consistently among the best in previous TREC's, and is again confirmed in TREC-6. As usual we ran the main tasks of ad-hoc and routing retrieval experiments. In addition, we also participated in the Chinese, batch filtering and the high precision tracks. It has been a busy and time-consuming endeavor as many of the ad-hoc and filtering experiments were done three times using different query lengths or to meet different objectives.

PIRCS has been described in previous TREC proceedings and references thereof. It is based on probabilistic indexing and retrieval, conceptualized as a three layer network with adaptive capability to support feedback and query expansion, and operates via activation spreading. The basic model evaluates a retrieval status value (RSV) for each query (q_a) document (d_i) pair as a combination of a query-focused process that spreads activation from document to query through common terms k, and an analogous document-focused process operating vice versa, as follows:

$$RSV = \alpha * \Sigma_k S(q_{ak}/L_a) * w_{ik} + (1-\alpha) * \Sigma_k S(d_{ik}/L_i) * w_{ak}$$

where $0 \le \alpha \le 1$, q_{ak} , d_{ik} are the frequency of term k in a query or document respectively, L_a, L_i are the query or document lengths, and S(.) is a sigmoid-like function to suppress outlying values. A major difference of our model from other probabilistic approaches is to treat a document or query as non-monolithic, but constituted of conceptual components (which we approximate as terms). This leads us to formulate in a collection of components rather than documents, and allows us to account for the non-binary occurrence of terms in a natural way. For example, in the usual discriminatory weighting formula for term k: w_{ak} = log [p*(1-q)/(1-p)/q], p = Pr(term k present | relevant) is set to a query 'self-learn' value of q_{ak}/L_a , and q = Pr(term k present |~relevant) = F_k/M, the collection term frequency of k, F_k, divided by the total number of terms M used in the collection. This we call the inverse collection term frequency ICTF and differs from the usual IDF. Moreover, as the system learns from relevant documents, p can be trained to a value intermediate between the basic self-learn value and that given by the known relevants according to a learning procedure [Kwok 1995]. Our system also uses two-word adjacency phrases as terms for

representation in addition to single words, and deals with long documents by segmenting them into approximately equal sub-documents of 550 words ending on a paragraph boundary. For the final retrieval list, retrieval status values (RSV) of the top three sub-documents of the same document are combined with decreasing weights to return a final RSV. This in effect favors retrieval of longer documents that contain positive evidence in different sub-parts of it, and differs from the approach that uses document length normalization weighting via training from previous documents [Singal, Buckley & Mitra 1996]. These PIRCS strategies have been in use since TREC-2.

The remaining parts of this paper are organized into four main sections describing the Chinese, ad-hoc and high precision, routing and filtering experiments, and our conclusion.

2. Chinese Track

2.1 Methodology

Chinese ad-hoc experiments consist of 26 new topics (number 29 -54) retrieving against the same 170 MB Xinhua and Peoples' Daily collections of last year. In an effort to study the effect of query lengths on retrieval effectiveness as in English ad-hoc, we perform three experiments using different portions of a given topic as query, namely: title only, description only (desc) and all sections (all). Our convention for naming these runs are: pirc7Ct, pirc7Cd and pirc7Ca respectively, with 7 denoting 1997 and C for Chinese.

We continue to use the short-word segmenter that we developed last year to segment Chinese texts. Short-word means words of 1 to 3 characters long. This procedure involves four steps: (i) lexicon look-up using longest match to segment input texts into smaller chunks; (ii) simple language rules to segment chunks into short-word candidates; (iii) discover new short-words based on frequency filtering; (iv) expand initial lexicon with the new short-words and re-process collection. Last year's initial lexicon of 2K has been enlarged to 27K entries to provide better coverage of common short-words. After adding new words discovered from the collection, the final lexicon size is about 43K. In addition, no stopwords are used as we have shown in [Kwok 1997b] that their effect on retrieval is minimal. On the other hand, accidentally removing a crucial stopword in particular queries may substantially bring down retrieval effectiveness, especially for short ones.

In [Kwok 1997a], we also have studied the behavior of retrieval using three representation types separately, viz: character, bigram and short-word with character. Retrieval using character representation alone is much inferior to the other two, because single characters are ambiguous. Bigrams are much more specific, and they exhaustively cover the Chinese words used in the collection that are two characters in size. However, they suffer from over-generation as one does not have good heuristics to decide which bigram is meaningful, and which ones are not. Moreover, some Chinese words are truly single character, and these would not be represented accurately with bigrams alone. Short-word representation in theory is the best, but can be in error because the segmentation algorithm may give wrong results. We recognize the problem with segmentation, and have used short-words together with characters simultaneously in a document or query for representation last year. Both bigrams and short-word indexing with characters provide similar retrieval effectiveness.

For the TREC-6 topics, using short-word indexing with character gives the following average query lengths: 6 for title, 11.6 for 'desc', and 37.8 for 'all'. With bigram indexing the corresponding figures of: 9.4, 10 and 65.2 respectively. We conjecture that bigrams and short-word indexing with characters may complement each other, and have used a combination strategy in these TREC-6 experiments. The collection was indexed two ways: bigram and short-word with character representation. For each query, two separate retrievals were performed using the two representations, and the resultant document lists are combined using equal weights. This combination strategy is employed for the long and short queries only (i.e. 'all' and 'desc' query types). For the very short queries, we use only short-word indexing, because bigram results for these titles can be quite poor compared to short-words. Two-stage retrieval with pseudo-feedback is employed as the standard retrieval strategy. For the short queries using title or description only, avtf weighting of query terms is employed {Kwok 1996]. In addition, the PIRCS system parameters are tuned as discussed in [Kwok 1997a].

2.2 Chinese Ad-Hoc Results

Official results for our Chinese retrieval are tabulated in Table 1, with three data columns for the very short (title), short (description) and long queries (all sections). Percentage changes are measured from values in the 'title' column. It appears that PIRCS continues to perform correctly for processing Chinese, and our strategy of combination pays off nicely. Thus, the average precision value of 0.6263 for query type 'all' is very high, and the 2795 number of relevants retrieved at 1000 documents represents over 94% of the 'pooled' known relevants of

2928. Precision at 10 documents of 0.8737 means close to 9 of the top 10 retrieved are relevant. Even at 100 documents retrieved, over 55% of them are on target. Using shorter queries of the description section only (desc) and title only leads to successfully less performance. However, the average precision of 0.4755 is still quite high for title queries having only on average of 6 terms. It appears that these topics are quite 'easy' for retrieval, as they often involve low level factual terms and few generalized concepts. Moreover, none of the queries has fewer than 16 relevant documents in the collection, and it is often these cases of just a few relevants that can lead to difficult retrievals with low precision values.

Query Type

	Title %	Desc %	All %
Relv.Ret	2547	2674 5	2795 10
Avg.Prec	.4755	.5423 14	.6263 32
P@10	.7115	.7962 12	.8737 23
P@20	.6692	.7519 12	.8135 22
P@30	.6192	.6974 13	.7718 25
P@100	.4327	.5035 16	.5542 28

Comparison with Median

	> = <	> = <	> = <
Avg.Prec	8 1 17,6	16,1 1 9	25,8 0 1
RR@100	6 2 18,6	15 3 8	21,1 3 2
RR@1K	10,3 7 9,1	13,2 8 5	18,6 6 2

Table 1: Chinese Ad Hoc Results for 26 Queries

In the same table, we also have the comparison with the medium from all sites. Under the query type 'all', the Avg.Prec row shows that 25 of our queries are better than medium, with 8 of them being best, and only 1 is below medium. Under other query types, the comparison is less favorable. For example, very short 'title' queries only have 8 above medium, 1 equal and 17 below medium; of these 17, 6 queries return the worst results. This is to be expected since there is only one standard for comparison: results from all query types are used to obtain the medium without differentiation.

As an example of the benefit of combination, we focus on the 'all' query type. The bigram retrieval composing this result has by itself an average precision of .5755 and relevants retrieved of 2735. Similarly, the short-word indexing with character alone has .6031 and 2791 values for these measures. They combine to give values of .6263 and 2795 respectively, an improvement of about 4% in precision from the better one, and a few more in the relevants retrieved.

In almost all cases, two-stage retrieval improves over one stage only, ranging from 0% to nearly 32% in the case of titles using short-word with character representation. Results for avtf weighting of short queries however are not uniform. For bigrams, improvements in all four cases of title and description during 1st and 2nd stage retrieval are observed. For short-word with character, only the first stage retrieval using title representation has slight improvment. Had we combined short-word without avtf retrieval and bigram with avtf, the average precision would have been .502 for the title (versus the official .476), and .566 for description only (versus .542).

For two years in a row Chinese retrieval in the TREC environment has shown much higher effectiveness (50-100% higher) than English for both long and short queries. It is not clear if this is due to the data being much 'easier', or if this is due to some intrinsic properties of the Chinese language. It is of interest to continue further experiments using more diverse collections and queries to throw some light into this phenomenon.

3. Ad-Hoc Retrieval and High Precision Track

3.1 Methodology

The ad-hoc task requires retrieval of 50 topics numbered 301-350 on the old TREC disk 4 consisting of Congressional Records and Federal Registry, and a new disk 5 of FBIS and LA Times documents. From the given topics, three types of queries can be formed according to what sections of the topics are used to define the queries: very short queries obtained from the title section only, short ones from the description section only (desc), and long queries from all sections (all). The average number of unique terms in each of these query types are: 2.58, 7.12 and 21.26 respectively. For five of these queries, there is only one word in the title query type. Except for one, these are highly specific terms: 'hydroponics', 'polygamy', 'retir', 'agoraphobia', 'metabolism'. Two of these five actually have more words in the title, but some are removed due to frequency considerations. Some of the other titles are also specific, containing two or three words.

We continue to use 2-stage retrieval with pseudo-feedback as in last year. For short queries we try out a document re-ranking procedure on the outputs of the initial retrieval. The purpose is to promote relevant documents to the top in order to enhance the quality of the 'feedback' documents and thereby improving second stage retrieval results. Re-

ranking is based on giving additional weights to selected query term pairs that appear in documents as further evidence of relevance. Short queries having few terms have the advantage that most probably all terms are important for relevance judgment, and selection of term pairs may be less prone to error. For longer queries, with numerous term pairs, selecting useful term pairs become more difficult and wrong ones may actually harm retrieval. We decided to use the procedure only on the very short queries from titles only. It turns out that the procedure depresses results slightly for both cases of short and very short queries.

3.2 Ad-Hoc Retrieval Results

Our official ad-hoc results are shown in Table 2 for the 3 query types. Most unexpected is the finding that average precision values for very short queries derived from 'titles' actually perform best with .2556, followed with the queries from 'all' sections with .2332. However, the quality of retrieval at the top 10 to 20 documents is better with the 'all' query type (e.g. P@10 = .4260 vs .4020), as well as the number of relevants retrieved at 1000 documents (Relv.Ret = 2674 vs 2384). 2674 is about 58% of the 'pooled' relevants of 4611. Also, the 'all' queries have much more favorable comparison with the medium from all sites (41 better or equal with 5 being best, and 9 below) than the 'title' queries (30 better or equal with 6 best, and 20 below with 3 worst). This rather erratic behavior we believe is due to the very good retrievals of a few very specific, single term title queries, as discussed below.

Query Types

	Title %	Desc %	All %
Relv.Ret	2377	1728 -27	2674 12
Avg.Prec	.2556	.1533 -40	.2332 -9
P@10	.4020	.2940 -27	.4260 6
P@20	.3390	.2450 -28	.3460 2
P@30	.3093	.2133 -31	.3093 0
P@100	.2092	.1330 -36	.2120 1

Comparison with Median

	> = <	> = <	> = <
Avg.Prec	26,6 4 20, 3	26,4 2 22	38,5 3 9
RR@100	25,10 12 13, 7	23,4 10 17,5	33,10 11 6
RR@1K	24,10 15 11,2	21,6 9 20	37,18 10 3

Table 2: Ad Hoc Results for 50 Queries

Our results for the 'desc' query type has been found to be erroneous due to some procedures being incompletely run for certain queries, and we did not realize it at the time of experiment and submission. Re-doing it properly gives an average precision of .1928 and relevants retrieved of 1938. Since some of the descriptions lack title terms, we also did an experiment that includes both the title and description section to form queries. The results are .2247 for precision and 2468 for relevants retrieved. The precision is still below that of the title query run, in contrast to some other reports.

A comparison of results among the query types shows that a single highly specific term can give very good results, and that the description section which tries to explain what the title word says in plain and less technical vocabulary (sometimes without the specific term) fails miserably. These plain terms are often of high frequency and ambiguous in meaning. The retrieval engine is not sufficiently intelligent to comprehend the explanation, gets confounded with all the common terms, and retrieve accordingly. Long queries also suffer from the noise introduced. In the following we show the average precision and relevants retrieved results of the five queries that have only one single term:

#312 Title: 'hydroponics'; description: 'the science of growing plants in water or some substance other than soil'.
#316 Title: 'polygamy'; description: 'a look at the roots and prevalence of polygamy, in the world today'.
#318 Title: 'retir'; description: 'aside from the united states, which country offers the best living conditions and quality of life for a u.s. retiree?'
348: Title: 'agoraphobia'; description: 'is the fear of open or public places(agoraphobia) a widespread disorder or relatively unknown?'.
#349 Title: 'metabolism'; description: 'the chemical reactions necessary to keep living cells healthy and/or producing energy'.

query	title	desc	desc*	all
#312	.9151/11	.0016/ 4	.0016/ 4	.1580/11
#316	.6926/26	.3914/25	.3914/25	.4158/25
#318	.0000/ 0	.0030/20	.0030/20	.0049/21
#348	.8100/ 5	.1411/ 3	.1436/ 5	.0528/ 3
#349	.2610/35	.0098/14	.0070/16	.2180/41

*corrected

Average Precision/Relevants Retrieved @1000

Except for #318 where all query types perform poorly, 'title' gives excellent to medium results. Just the difference of these five between 'title' and 'all' adds 0.0366 to the 'all' average precision result, bringing it to 0.2698 for the latter. Similar large differences can be accounted for when comparing 'title' and 'desc' results.

In contrast, the description section of the Chinese queries actually contains related concept terms and are useful for retrieval, giving 14% improvement in average precision when compared with 'title' only queries.

3.3 High Precision Track

Initially we intend to participate in this track via manual intervention. Due to time constraints, we found that we could not make the deadline and decided to submit the top documents of our ad-hoc runs. Thus, our high precision results are fully automatic without manual handling. This in effect would provide a lower-bound to this track. As a method to boost precision, we in addition ran our re-ranking procedure described earlier on the final retrieval lists for the 'title' query type. This turns out to depress its value a little bit.

Results are tabulated in Table 3. They are not competitive and far below medium because we lack a human feedback loop. In the comparison of precision @10 with medium, it is interesting to note that there are always some queries (8 for 'title', 1 for 'desc' and 4 for 'all') for which our automatic method can achieve maximum without manual help. They include all four of the highly specific single term queries.

Query Types

	Title %	Desc %	All %
P@10	.3980	.3360 -16	.4260 7
RP@10	.4163	.3509 -16	.4384 5
unrP@10	.0766	.0561 -27	.0574 -25

Comparison with Median

	> = <	> = <	> = <
P@10	10,8 10 30, 19	7,1 12 31, 15	8,4 13 29, 6

Table 3: High Precision Results for 50 Queries

4. Routing Retrieval and Filtering track

4.1 Methodology

4.1.1 Combination of Retrievals

Combination of retrievals have been studied by us as well as other groups (see e.g. [Hull, Pedersen & Schutze 1996] and references thereof). Two different retrievals using different methods, added together will frequently be superior to both of them individually. One reason is that non-relevant documents tend to be more random in the retrieved pool. Another reason for adding retrievals is, that no method performs best in every situation. It may be easier to build small specialist queries and add them after retrieval, favoring the better performer.

A simple specialist query is one developed from the text of the original query. It will perform well compared to a query that is based on relevant documents, if the available relevant documents do not cover all the facets of the query concept, or if the documents contain mainly irrelevant information.

Many of the queries have training data available for as far back as 1987. It is a concern, that the old information is not timely anymore, and will in fact harm the query. A specialist query trained from the current FBIS5 documents only will avoid this problem.

4.1.2 Routing Query Pirc7R1

The first routing query submitted is an addition of four retrievals. All of then use the PIRCS retrieval engine, the training documents and the dictionary used are different.

a) Retrieval with no training documents.

This is essentially the same retrieval as used for ad-hoc. This retrieval will performs well if the training documents are noisy or if the original query can be defined by a few well chosen terms. Examples where it did well are query 10003 "Privatization in Peru" and query 78 "Greenpeace activities".

b) Retrieval using the FBIS5 training documents only.

This retrieval will perform well if the query concept is time dependent. Examples where it did well are query 228 "Success stories in recent years concerning environmental recovery", query 100 "Controlling transfer of high technology" and query 1 " A Pending antitrust case".

These two retrievals use only statistics available from the FBIS5 collection and their combination comprised the retrieval used for the filtering track

c) Two retrievals using genetic algorithms to select training documents.

Many queries have a large number of evaluated training documents of varying quality. Retrieval can be improved if only a subset of training documents are used. We use genetic algorithms search to find this subset for the 39 queries that were repeated from TREC5. (For the others we used the top sub-documents for each relevant document as usual).

Genetic algorithms [Gol89] is a search procedure based on the survival of the fittest. The algorithm can be summarized as follows [Gre88]

```
Procedure GA
begin
        initialize population P(0)
        evaluate P(0)
        t=1
        repeat
                select P(t) from P(t-1)
                recombine P(t)
                evaluate P(t)
        until (termination condition)
end
```

We use genetic algorithms in this search space by performing the following steps.

1. Create training and testing query region. The top 1500 retrieved documents from the fbis 5 collection plus all short judged documents from the other collections are divided into 2 equal parts for this purpose.
2. Run GA for 6 generations to select best query
3. Swap training and testing data
4. Run GA for 6 generations to select best query

This procedure yields two more retrievals.

4.1.3 Routing Query Pirc7R2

We have recently began experimenting with utilizing backpropagation neural network for information retrieval. Our starting point was NevProp, a publicly available c program maintained by Phil Goodman of the University of Nevada. The results by itself were not good, but it invariably enhance the pircs retrieval, when the two retrievals were combined.

The input layer of the backprop neural network consists of about 100 terms selected by the pircs system, plus 60 positive term pairs plus 6 negative term pairs. The hidden layer has 3 nodes and the output layer has one node. This parameters were selected after experimentation with a very limited number of queries. The training data was the same as the training data for the genetic algorithms search.

The second routing query adds a retrieval by a backpropagation neural network to the first routing query. While the bp retrieval by itself was not as good as pircs, it is sufficiently different from it to make addition beneficial.

4.1.4 Performance based addition of retrievals.

Retrievals are combined using the equation

$$RSV_j = \sum_i C_i * RSV_{ji}$$

The RSV of the j-th document is the sum of the RSVs for all retrievals I, multiplied by a constant Ci for retrieval I. We may attempt to predict the performance of a method by doing a retrieval on the training data. In general the overall performance is better if a higher value is assigned to the better retrieval.

For trec4 we experimented with setting Ci to the 11 point average and normalizing retrieval weights by dividing all RSVs by the highest ranked RSV. There are some problems with this method. Although the 11pt average is an indication of the strength of the retrieval, it may not be the most optimal coefficient. The normalization also presents a problem. Bad retrievals frequently have a flat RSV curve, the normalization procedure would result with all documents RSV being close to 1.0000.

To overcome these problems, we do not normalize and we determine the Ci by trying to maximize the 11pt average on the training database.

4.2 Routing Retrieval Results

	pircs7R1			pircs7R2		
	>	=	<	>	=	<
avg prec	36(2)	1	10	36(6)	3	8
rel ret @100	34(6)	4	9	36(6)	3	8
rel ret @1000	34(12)	6(3)	7	34(12)	7(3)	6

Table 4.1 Comparison of routing results with mean. Number in parenthesis is number of best values.

Performance of the official runs was well above median. (see table 4.1) . The strategy of adding retrievals appears to have paid off.

Looking at individual runs (Table 4.2) further underscores the value of combining specialist retrievals. The backprop run although weak by itself improved the pircs7R1 resulting in pircs7R2 our best result, similarly the query only run (no 1) when added to the fbis only run (no2) resulted in the filtering run, an over 10% improvement.

Table 4.3 shows the performance of various retrievals on the FBIS5 collection. As can be seen the run which only used the FBIS training documents performed much better than all the others. This resulted in inflated values for the addition coefficients for this run at the combination phase. Although we limited the maximum coefficient to .6, further limits should have been place on this run, since it was purely retrospective as opposed to the others which used all the training data.

run no	name	avg	pct imp	rel retr	pct imp
1	query only	0.2487	0%	4648	0%
2	fbis only	0.2992	20%	5120	10%
3	all docs	0.3114	25%	4793	3%
4	ga 1	0.3362	35%	5082	9%
5	ga 2	0.3431	38%	5087	9%
6	backprop	0.2417	-3%	4611	-1%
add runs					
1+2	filtering	0.3325	34%	5109	10%
4+5	ga	0.3461	39%	5097	10%
4+5+6	ga+np	0.3501	41%	5124	10%
1+2+4+5	pircs7R1	0.3605	45%	5284	14%
1+2+4+5+6	pircs7R2	0.3783	52%	5288	14%

Table 4.2 Routing runs and combinations on FBIS6.

run no	name	avg	pct imp
1	query only	0.2199	0%
2	fbis only	0.6231	183%
3	all docs	0.3831	74%
4	ga 1	0.4220	92%
5	ga 2	0.4145	88%
6	backprop	0.3297	50%
add runs			
1+2	filtering	0.6389	191%
4+5	ga	0.4336	97%
4+5+6	ga+np	0.4780	117%

Table 4.3 Routing runs and combinations on FBIS5.

Since the ga training was only run for the 38 old queries, we separated performance statistics into old and new queries in table 4.4. The gain over the 38 old queries for the ga run (add 4+5) was 12% over run 3 which is the pircs run using all top relevant subdocuments. Another interesting detail is that for the 9 new queries, there is very little gain made by training from relevant documents, but combination of retrievals averted disaster. Comparing runs 2 and 3 shows the importance of term statistics. All runs except for runs 1 and 2 used the old trec5 dictionary and statistics. For the old 38 queries the run 2 using all training docs did much better than run 3, which had access only to the fbis 5 training docs, for the new 9 where the training documents were the same, run 3 did much better. The decision not to include the additional words and statistics from fbis5, which was made for time saving reasons, probably hurt overall performance.

run no	name	avg old 38	pct imp	avg new 9	pct imp
1	query only	0.2422	0%	0.2765	0%
2	fbis only	0.2943	22%	0.3012	9%
3	all docs	0.3341	38%	0.2156	-22%
4	ga 1	0.3648	51%		
5	ga 2	0.3733	54%		
6	backprop	0.2664	10%	0.1376	-50%
add runs					
4+5	ga	0.3771	56%		
**1+2+4+5	pircs7R1	0.3678	52%	0.3294	19%
**1+2+4+5+6	pircs7R2	0.3921	62%	0.3201	16%

Table 4.4 Routing runs and combinations on fbis 6 separated into old and new queries.
** note: for the avg new 9 col the runs don not include runs 4 and 5

4.3 Filtering Track

The filtering query is a standard pircs query based on the fbis 5 documents only, added to a query that is based on the original text of the query. Thresholds were developed by finding the rsv for which the maximum value is reached for each of the three utility measures. The second set lowers the threshold by 10%, based on the observation that if the retrieval is retrospective the relevant documents have a lower rank and the maximum rsv is reached earlier.

run	>	=	<
pirc7f11	15(4)	12(2)	20
pirc7f12	17(3)	13(2)	17
pirc7f21	19(4)	4	24
pirc7f22	23(5)	5	19
pirc7a1	16(1)	15	16
pirc7a2	23(1)	14	10

Table 4.5 Comparison of filtering results with mean. Number in parenthesis is number of best values.

We report here on the corrected retrieval submitted after the trec conference. The results for f1 and f2 utility values are a bit less then the median, while for the asp utility value is above. Lowering the thresholds improved the asp run, but made the f1 and f2 runs worse.

5. Conclusion

TREC-6 Chinese experiments continue to confirm TREC-5 observations that accurate word segmentation is not a pre-requisite for effective IR. Using a combination of simple representations, our PIRCS engine is able to return the best and very high effectiveness for another set of 26 queries. Our ad-hoc English retrieval is also among the best two submissions for both very short and long queries. Single word queries of a very specific and unambiguous nature performs like database retrieval and can give very good results. Longer descriptions only serve to confuse the retrieval engine under such circumstances. However, most information needs cannot be described by this type of representation. Our routing experiments introduce combination of new retrieval methods and also perform admirably among the top four submissions. It appears that PIRCS can perform consistently good under various circumstances.

Acknowledgments

This work is partially supported by a grant from the Department of Defense.

References

Goldberg, D. E. Genetic Algorithms in Search Optimization & Machne Learning. Addison Wesley 1989.

Hull, D.A., Pedersen, J.O. & Schutze, H (1996). Method combination for document retrieval. In: Proc. 19th Ann. Intl. ACM SIGIR Conf. On R&D in IR. Frei, H.P., Harman, D. Schauble, P. & Wilkinson, R. Aug 18-22, 1996. Pp.279-287.

Kwok, K.L. (1997a). Comparing representations in Chinese information retrieval. In: Proc. 20th Ann. Intl. ACM SIGIR Conf. On R&D in IR. Belkin, N.J., Narasimhalu, D. & Willett, P. (eds). July 27-31, 1997. Pp.34-41.

Kwok, K.L. (1997b). Lexicon effects on Chinese information retrieval. In: Proc. 2nd Conf. On Empirical Methods in NLP. Cardie, C. & Weischedel, R. (eds). Aug 1-2, 1997. Pp.141-148.

Kwok, K.L. (1996). A new method of weighting query terms for ad-hoc retrieval. In: Proc. 19th Ann. Intl. ACM SIGIR Conf. On R&D in IR. Frei, H.P., Harman, D. Schauble, P. & Wilkinson, R. Aug 18-22, 1996. Pp.187-195.

Kwok, K.L. (1995). A network approach to probabilistic information retrieval. ACM Transactions on Office Information Systems. 13:325-353.

Kwok, K.L. & Grunfeld, L (1997). TREC-5 English and Chinese experiments using PIRCS. In: The Fifth Text REtrieval Conference (TREC-5). To be published.

Kwok, K.L. & Grunfeld, L (1996). TREC-4 ad-hoc, routing retrieval and filtering experiments using PIRCS. In: The Fourth Text REtrieval Conference (TREC-4). Harman, D.K. (Ed.). NIST Special Publication 500-236, pp.145-152.

Singhal, A., Buckley, C. & Mitra, M (1996). Pivoted document length normalization. In: Proc. 19th Ann. Intl. ACM SIGIR Conf. On R&D in IR. Frei, H.P., Harman, D. Schauble, P. & Wilkinson, R. Aug 18-22, 1996. Pp.21-29.

AT&T at TREC-6

Amit Singhal
AT&T Labs–Research
singhal@research.att.com

Abstract

TREC-6 is AT&T's first independent TREC participation. We are participating in the main tasks (adhoc, routing), the filtering track, the VLC track, and the SDR track[1] This year, in the main tasks, we experimented with multi-pass query expansion using Rocchio's formulation. We concentrated a reasonable amount of our effort on our VLC track system, which is based on locally distributed, disjoint, and smaller sub-collections of the large collection. Our filtering track runs are based on our routing runs, followed by similarity thresholding to make a binary decision of the relevance prediction for a document.

1 Introduction

TREC-6 is the first TREC in which AT&T is participating as an independent group. Much of our work is largely inspired by Smart's philosophy of fully automatic processing of large text collections. Our participation is based on an internally modified version of Cornell's SMART system. We submitted runs for the adhoc task, the routing task, the filtering track, the VLC track, and the SDR track (see footnote 1).

In the main tasks, the highlight of our preparation for TREC this year was our repeated failure to improve upon Cornell's TREC-5 results (we were a part of Cornell's TREC participation last year). In the routing task, we tried many new techniques, and variations of old techniques, but nothing provided a noticeable improvement in performance over last year's results. We finally settled for a two-pass query modification algorithm, with the second pass intended to fix the weakness of the first-pass query. This yields small improvements in our routing performance. In the adhoc task, we augment the "goodness" of a query-term by a new "importance factor" in addition to the usual query term weight, for selecting the top documents to be used in pseudo-feedback.

2 Routing Runs

Our routing runs use routing queries learned in a query zone using Rocchio's formulation. [7] All term weighting in our system is based on pivoted-unique document length normalization. [6] The first official run, **att97rc** (routing, conservative), uses the routing algorithm presented in Table 2.

Unfortunately, **our official run att97rc has a bug** that resulted in non-optimized word-pair weights. Fixing it improves the performance reasonably. Table 3 shows the results of our buggy official run **att97rc**, as well as the results when the bug is fixed. Since the fixed run was not in the pool of runs used to compute the best/median statistics, we notice that the fixed run is actually better than the best official result for three of the topics, and is above median for 46 out of 47 topics. These numbers suggest that the above routing algorithm is quite effective.

Table 4 shows the effectiveness of various components of the above routing algorithm. When no query zoning is used, *i.e.*, all non-relevant articles are used in Rocchio's formula, a different set of Rocchio parameters ($\alpha = 8$, $\beta = 64$, $\gamma = 256$) is known to be more effective [7], and we obtain an average precision of 0.3296. Once we switch to using query zones, we obtain a 8% improvement over not using query zones. This is in strong agreement with our earlier experiments on other TREC routing tasks. [7] Now we can either optimize

[1] This report does not describe our SDR track participation. Please see the adjoining report "AT&T at TREC-6: SDR Track" for details of our SDR system.

l tf factor:

$$1 + log(tf)$$

L tf factor:

$$\frac{1 + log(tf)}{1 + log(average\ tf\ in\ text)}$$

t idf factor:

$$log(\frac{N+1}{df})$$

u length normalization factor:

$$\frac{1}{0.8 + 0.2 \times \frac{number\ of\ unique\ words\ in\ text}{average\ number\ of\ unique\ words\ per\ document}}$$

where, *tf* is the term's frequency in text (query/document)
N is the total number of documents in the training collection
df is the number of documents that contain the term, and
the average number of words per document is 110.

ltu weighting: **l** factor × **t** factor × **u** factor
Lnu weighting: **L** factor × **u** factor
Ltu weighting: **L** factor × **t** factor × **u** factor

Table 1: Term Weighting Schemes

the query without adding word-pairs, or after adding word-pairs. If we optimize the query without adding word-pairs, we get an overall improvement of about 16% over our baseline. But if we do add word-pairs (as explained in step 3 of the algorithm in Table 2), prior to optimization, just by adding 100 pairs, we get an improvement of about 13% over the baseline. Optimization of pair-added queries yields even richer improvements than optimizing the non-pair-added queries, yielding an overall improvement of about 25% over our baseline.

The above routing algorithm is quite similar to the routing algorithm we used in TREC-5 [2], except for minor variations. We tried various new techniques to improve upon the above routing algorithm, but none of the techniques we tried yielded better results that the above algorithm.

Our first approach revolved around clustering the known relevant articles for a query. The main thought behind this approach was that relevance can have "multiple aspects". For example, for a query on *trade barriers in Japan*, one aspect of the relevant documents is trade barriers in the automobile industry (with keywords like: *chrysler, ford, mitsubishi, ...*), yet another aspect is trade barriers in the electronics industry (with keywords like: *toshiba, sony, ...*). If one can isolate such patterns in the relevant documents, it should be possible to learn one query per aspect and this query should be better than one global query for routing documents related to that aspect. Unfortunately we were unable to improve our routing performance using such an approach, mainly, we believe, due to the following reasons: a) not too many queries have clearly defined multiple aspects of relevance, b) once we cluster documents and select an aspect, the amount of training data (relevant and non-relevant documents) is much less for the aspect, resulting is a poorer feedback query; and c) a good single query already incorporates the multiple aspects of relevance in it, for example, the feedback query for the above example will have keywords from all aspects (*chrysler, ford, mitsubishi, toshiba, sony, ...*), thereby implicitly giving us the benefits we had hoped to obtain from clustering.

The second approach we tried was based on using a multi-pass query refinement technique. The basic idea behind this scheme is to compensate for the deficiency of a feedback-query, by enhancing it with another pass of feedback. For example, once we learn a first pass query using Rocchio's formulation (no optimization), we can use this feedback query to rank the *training* collection. This feedback query will rank some non-relevant documents at top ranks. These are the non-relevant documents that the first pass feedback-query is having difficulty "defeating". If we learn another query specifically aimed at defeating these non-relevant documents

1. Using *ltu* weighted queries (see Table 1), and *Lnu* weighted *training documents*, form a training "query-zone" by retrieving the top 5,000 documents for the query (using the inner-product similarity).

2. Using the non-relevant documents in the query-zone, and *all* the relevant documents in the training corpus, form a feedback query using Rocchio's formulation using the following constraints/parameters:

   - Document terms are *Ltu* weighted. Original queries are *ltu* weighted.
   - Only the original query terms, and the "non-random" words and phrases, *i.e.*, the words that appear in at least 10% of the relevant articles, and phrases that occur in at least 5% of the relevant articles are considered for use in the feedback query.
   - Top 100 words and 20 phrases, as weighted by the Rocchio formula:

     $$8 \times \textit{original query vector} + 64 \times \textit{average relevant vector} - 64 \times \textit{average nonrelevant vector}$$

     are retained in the feedback query with weights predicted by the above formula. The average relevant vector is the average vector of all the relevant documents: $\frac{1}{|R|} \times \sum_{D_i \in Rel} \vec{D}_i$, where $|R|$ is the number of known relevant documents. The average non-relevant vector is defined correspondingly.

3. The query formed in the above step is a recall-oriented query. To enhance the precision of the query, we add query-word—query-word cooccurrence pairs to the above query. If two words occur *in the same document*, they form a potential cooccurrence pair.

   - Using the 100 query words from the previous step, we consider the 4,950 word-pairs.
   - All the "random" word-pairs, *i.e.*, the word-pairs that occur in fewer than 7% of the relevant documents, are removed.
   - Since we want to add a precision tool to the query, we re-sample the training non-relevant documents, and use the *top* $2 \times |R|$ non-relevant documents from step 1. Here $|R|$ is the number of training relevant documents.
   - Using *all* the relevant documents, and this restricted set of non-relevant documents (a tighter query-zone, so to speak), we add to the query (from step 2) the 100 word-pairs with highest weights as weighted by the following Rocchio formula:

     $$64 \times \textit{average relevant vector} - 64 \times \textit{average nonrelevant vector}$$

   Since word-pair weights in documents are needed in the above formula, to compute the *Ltu* weight for a pair, the lower of the constituent words' *tf* is considered as the pair's *tf*. Pair *idf* is computed on the fly by computing the true pair *df* by intersecting the individual words' inverted lists.

4. Term weights in this query of 100 words, 20 phrases and 100 word-pairs are further optimized using three-pass dynamic feedback optimization (DFO) with pass ratios 1.00, 0.50, and 0.25. [1]

5. The optimized feedback query is used to rank the new (test) documents. The test documents are *Lnu* weighted (see Table 1).

Table 2: Routing Algorithm

Run	Average Precision	> Best	Best	>= Median	< Median
Official (buggy)	0.3963	–	4	43	4
Fixed	0.4132	3	0	46	1

Table 3: Results for **att97rc**

	No QZ $\alpha.\beta.\gamma$: 8.64.256	QZ $\alpha.\beta.\gamma$: 8.64.64	QZ+DFO (No Pairs)	QZ+Pairs (No DFO)	QZ+Pairs+DFO
Avg. Prec	0.3296	0.3560	0.3819	0.3716	0.4132
Improvement (over No QZ)	–	+ 8.0%	+15.9%	+12.7%	+25.4%

Table 4: Effect of various components of **att97rc**

Run	Average Precision	> Best	Best	>= Median	< Median
Official (buggy)	0.4207	–	4	45	2
Fixed	0.4307	3	0	45	2

Table 5: Results for **att97re**

(using Rocchio's formulation with all the relevant documents and these top few non-relevant documents), then by combining the first pass and the second pass query, we should be able to get an overall improved query. We found that such two-pass approach improves routing effectiveness in experiments on the TREC-3, 4, and 5 routing tasks, over using a single pass non-optimized feedback. But the resulting two pass query is still somewhat poorer than the optimized one pass query. Optimizing the two pass query didn't buy us much. Overall it is a wash to use a multi-pass query or a single pass optimized query.

A minor variation of the above multi-pass approach did yield very small improvements over an optimized one pass query for the TREC-3, 4, and 5 tasks, and was submitted as our other official run **att97re** (routing, experimental). The idea in this run is to find the relevant documents that the first pass feedback-query is not ranking well in the training collection, *i.e.*, the bottom ranked training relevant documents (as ranked by the feedback-query), and the non-relevant documents that the first pass query is not defeating well, *i.e.*, the top ranked training non-relevant documents. This idea bears resemblance to the class of algorithms known as boosting in the machine learning community. [3] We select the bottom $|R|/2$ relevant documents, and the top $2 \times |R|$ non-relevant documents (where $|R|$ is the number of training relevant documents for a query).

We take the query formed using steps 1-3 of the algorithm in Table 2, rank the training collection using this query, and select the bottom $|R|/2$ relevant documents, and the top $2 \times |R|$ non-relevant documents. We independently form another query of 100 words, 20 phrases, and 100 word-pairs using these training documents (using steps 2-3 of the algorithm in Table 2). The final query is constructed using the following formula: pass-1 query + 0.25×pass-2 query. This final query is then optimized using a 3-pass DFO (as in step 4 in algorithm in Table 2). Unfortunately our official submission **att97re also has a bug**. The phrase and cooccurrence contributions were reduced (0.5 used in place of 1.0) due to a bug in the shell script used in **att97re**. Once the bug is fixed, the average precision for **att97re** improves some. This run is about 4% better than our conservative run **att97rc**. Table 5 shows the results of **att97re**. We believe that such multi-pass approaches for routing are promising, and deserve a more careful study.

Aside

In doing some post hoc analysis of where our current routing algorithms are failing, and why aren't we observing any marked improvements in the best routing effectiveness over the last few TRECs, we read several documents retrieved at top ranks by our routing queries. While reading through these documents, we did find many instances where a non-relevant article was ranked high because of the limitations of the statistical nature of our systems. But often enough, we found ourselves wondering why a document was judged non-relevant while another, very similar document was judged relevant. For the adhoc task, on reading the documents, it was much more obvious to us why documents were judged relevant or non-relevant. Voorhees and Harman report a three-way assessor agreement rate of approximately 72% for the adhoc task, [8] which is a very respectable agreement rate. We wonder if this figure would be lower for the routing task. It would be interesting to do such an assessor agreement study for the routing task, especially since the documents in the judgment pool are being retrieved by queries that have been learned using a large amount of training data, and are therefore much more precise (or effective) than the adhoc queries.

Word	df in 1,000	df	$\frac{df\ in\ 1,000}{df}$	Weight	Factor	Final Weight
hazard(ous)	386	7125	0.0542	5.34903	1.0000	5.34903
termin(als)	444	11903	0.0373	4.71901	0.6838	3.22673
comput(er)	454	22505	0.0202	3.93704	0.5528	2.17634
health	561	43015	0.0130	3.14174	0.4523	1.42094
daily	262	21034	0.0125	4.02003	0.3675	1.47754
individual(s)	474	44335	0.0107	3.10463	0.2929	0.90933
basi(s)	427	42023	0.0102	3.17038	0.2254	0.71461
work	617	148250	0.0042	1.62267	0.1633	0.26505

Table 6: Term Ordering for Topic 350

3 Adhoc Runs

Over the last few years, it has been shown that pseudo-feedback, *i.e.*, query modification without any relevance feedback from a user, *assuming* that the top few documents retrieved by the user query are relevant, yields noticeable improvements in retrieval effectiveness in the adhoc task. [4, 8] Typically we have been using the top twenty documents retrieved by the original query for pseudo-feedback. Motivated by Hearst's observations in [5], recently we have tried improving the quality of our relevance assumption by reranking the top fifty documents retrieved by the original query according to some "precision criteria" and using the top twenty documents from this reranked list in pseudo-feedback. [2] One particular criteria that we have used is the presence of several query terms in a small window of text in a document (see Table 7).

This year, we used a new method to rerank the top fifty documents to select the set of twenty documents used in pseudo-feedback. This technique is based on a new query term weight modification factor that we use to assess the importance of a query term in addition to the regular query term weight *ltu* (see Table 1). During experimentation, we observed that the goodness of a query term is related to the number of documents, in the top (say, 1,000) documents retrieved by the query, that contain the term. But since common words can appear in many documents, we need to normalize the above measure by the global df of the term. We used the following function to rank the original query terms:

$$\frac{number\ of\ documents\ in\ the\ top\ 1,000\ documents\ (retrieved\ by\ the\ query)\ that\ contain\ the\ term}{number\ of\ documents\ in\ the\ collection\ that\ contain\ the\ term\ (df)}$$

For example, for query 350, "Is it hazardous to the health of individuals to work with computer terminals on a daily basis?", the term ordering generated by this scheme is shown in Table 6. The query words are listed in decreasing order of their perceived importance in Table 6. This method does rank terms that we intuitively know are most important (*e.g.*, hazard, terminal) ahead of other terms that we think are less important (*e.g.*, basis, work). Even though a purely idf based ranking will place a relatively less useful word, like *basis*, ahead of a more useful word, like *health*.

After the terms in a query are ranked by the above formula, their weights are modified by multiplying with the following importance factor:

$$1.0 - \sqrt{\frac{rank - 1}{10}}$$

This factor lowers the weights of the terms ranked poorly in the above ranking, thereby emphasizing the top few terms noticeably. Table 6 also shows the original query term weight (column 5), the value of the above factor for a term (column 6), and the final query term weight for reranking the top fifty documents (column 7). We can see how less important terms, like *basis and work*, get a very low final weight. By using this weight modification factor, we ensure that a combination of the low ranked (hopefully less useful) terms will not defeat a presence of a high ranked term, which is often essential for relevance. Table 7 shows our full adhoc algorithm.

Table 8 show the performance of the various components of our adhoc algorithm over several TREC tasks. We use only the **description** field of the queries for the results reported in Table 8. The second column has the results for a straight vector run. The third column shows the results when the top 20 documents

1. Retrieve 1,000 documents using *ltu* weighted queries and *Lnu* weighted documents.
2. Rerank the query terms by $\frac{df \text{ in } 1,000}{\text{df}}$ and multiply their weight by $1.0 - \sqrt{\frac{rank-1}{10}}$.
3. Using the re-weighted query, rerank the top 50 documents. Documents are broken into 50 words overlapping windows starting at every 25th word, and a document's score is the best score of any window in that document.
4. Top 20 documents in this reranked list are *assumed* relevant. Since majority of the bottom ranked documents are usually non-relevant, documents ranked 501 to 1,000 are *assumed* to be non-relevant. Pseudo-feedback is performed using these assumptions, and the query is expanded by 25 words and 5 phrases. (Rocchio parameter values of $\alpha = 8$, $\beta = 8$, $\gamma = 8$ are used.)
5. The expanded query is used to rank the collection to get the final ranking for documents.

Table 7: Adhoc Algorithm

Task	No Feedback (Lnu.ltu)	Top 20 Relevant	501-1,000 Non-Relevant	Rerank based on locality
TREC-3	0.2385	0.3214	0.3340	0.3462
	–	+34.8%	+40.1%	+45.2%
TREC-4	0.2303	0.3000	0.3082	0.3167
	–	+30.2%	+33.8%	+37.5%
TREC-5	0.1505	0.1855	0.1909	0.2010
	–	+23.3%	+26.8%	+33.5%
TREC-6	0.1621	0.1723	0.1849	**0.1847**
	–	+ 6.3%	+14.1%	+14.0%

Table 8: Effect of various pseudo-feedback methods on adhoc performance, description-only queries. Our official run **att97ac** is shown in bold.

Task	No Feedback (Lnu.ltu)	Top 20 Relevant	501-1,000 Non-Relevant	Rerank based on locality
TREC-6 Title-Only Queries	0.2005 –	0.2017 + 0.6%	0.2079 + 3.7%	**0.2289** +14.2%
P@20	0.3070 –	0.3200 + 4.2%	0.3210 +4.6%	0.3530 + 15.0%

Table 9: Effect of various pseudo-feedback methods on title-only TREC-6 adhoc queries. Our official run **att97as** is shown in bold.

Query	Query Length	No Feedback (Lnu.ltu)	Top 20 Relevant	501-1,000 Non-Relevant	Rerank based on locality
Title Only (T)	3.02	0.2005	0.2017 (+ 0.6%)	0.2079 (+ 3.7%)	0.2289 (+14.2%)
Title+Desc (T+D) Improvement over T	11.78	0.2064 + 3.0%	0.1931 (– 6.5%) – 4.3%	0.2071 (+ 0.3%) – 0.4%	0.2237 (+ 8.4%) – 2.3%
Full (T+D+N) Improvement over T	33.60	0.2179 + 8.7%	0.2210 (+ 1.4%) + 9.6%	0.2282 (+ 4.7%) + 9.8%	0.2384 (+ 9.4%) + 4.1%

Table 10: Performance of different lengths of TREC-6 adhoc topics.

from the straight vector run are assumed to be relevant and pseudo-feedback is performed. The fourth column also assumes documents ranked 501-1,000 as non-relevant (in addition to the third column). The fifth column is the reranking run (in addition to assuming 501-1,000 non-relevant). It is evident that pseudo-feedback improves performance across tasks. However, we should note that the improvements obtained for this year's task are much lower than what we have been getting in the past (only 6% over a poor baseline vs. 23-35% over reasonable baselines). We also observe that assuming the bottom ranked documents to be non-relevant gives us some additional improvement in performance; actually an important 7-8% (over not assuming non-relevance) for this year's task. Also our reranking of the top documents to select a new set of twenty documents for feedback also gives us additional improvement across tasks, *except for this year's task.*

We believe that locality based reranking of top documents to select a better set of *assumed relevant* documents is a promising way to improve the quality of pseudo-expansion, but it needs more careful investigation. Since we developed this reranking scheme in the final days before the submission, we did not study various other alternatives that can be used for reranking in place of the above method. Also, the formula used above to marginalize the less important words was developed at the last moment and we believe that there are better ways of emphasizing core query terms than the adhoc formula we have used above.

Title-Only Queries

We also submitted a run for the very short, title-only queries. Our main motivation for this run was to test the robustness of our algorithms for these very short queries (which are very common these days in a web-search type environment). Table 9 shows the effect of various components of our adhoc algorithm on retrieval using these very short queries. For this task, pseudo-feedback doesn't yield much better results over basic vector matching, but pseudo feedback with reranking does yield about 14% improvement. This indicates that document reranking (for pseudo-feedback) is quite useful even for these tiny queries. For a casual searcher precision at twenty is usually a more meaningful number than average precision. Table 9 also shows the P@20 figures. We again see that reranking based pseudo-feedback gets (on an average) about one extra relevant document in the top 20 documents as compared to basic vector matching.

Query Length

We also study the effect of using longer user queries in adhoc searching. Table 10 shows the results of using the title-only queries, title+description queries, and title+description+narrative queries for this year's adhoc task. This scenario is akin to when a user progressively fleshes-out the query by further describing

Run	Average Precision	Best	>= Median	< Median
att97ac (desc only)	0.1847	1	36	14
att97ae (desc only)	0.1801	3	33	17
att97as (title only)	0.2289	7	31	19

Table 11: Results for adhoc runs

his/her information need to a system. The query length in column 2 is the average number of unique words and phrases in the query. In adding the description section to the title-only query, a user adds almost another nine new words and phrases to a query (the average query length increases from 3 to almost 12). In further adding the narrative section, the user adds an average of another twenty-two new words and phrases to the query (average query becomes 33.6).

A casual user will seldom provide a system with such (33 word) long queries. However, the good news is that with 3 carefully chosen words, the retrieval effectiveness of a query using our reranking-based algorithm is almost as good as the retrieval effectiveness of a very long query. Here are some key observations from Tables 8 and 10:

- For this year's adhoc task, just assuming that the top few documents retrieved by the initial query are relevant and doing relevance feedback is not very useful. This technique has been quite successful in the past. This year, depending on what parts of a topic are used in the initial run, this techniques loses or gains up to 6% in average precision. In the past, this feedback method has yielded large improvements (see Table 8, TREC-3–5 rows).
- Assuming that documents ranked poorly by the initial query are non-relevant does help some consistently. An exception is this year's description-only query (see Table 8) for which this assumption helps noticeably. This might be due to the poor baseline, or due to some other reason. We haven't investigated this yet. But, in general, there is no harm in using this assumption.
- Reranking the top few documents based on query-word locality to improve our assumption of relevance is quite useful in general. Except for the description-only queries for this year's task, this technique consistently yields improvements over no reranking. Once again, using this technique is seldom hurtful.
- Even though adding the description section to the queries somewhat improves the the initial queries, post pseudo-feedback, it is not very useful. (The improvements obtained over using the title-only queries are listed in the rows labeled *Improvement over T*.)
- Even though full queries are about 9% better than the title-only queries initially, (Table 10, column 3), post reranking pseudo-feedback, the results are just 4% better than the title-only results (column 6). This result is encouraging for locality-based reranking, since the performance gap between the very short and the very long queries reduces post reranking and pseudo-feedback.

Experimental Run

Our experimental run **att97ae** was based on the following reasoning: since pseudo-feedback is usually useful, if we do another pass of pseudo-feedback assuming that the first pass query is retrieving more relevant documents in the top ranks, and is pushing down more non-relevant documents to ranks 501-1000, we should be able to improve the results further. This half-hearted attempt didn't prove beneficial. Our experimental run yields poorer results than our first run—att97ac.

Table 11 gives comparison to medians for our submissions. Based on the number of queries for which we have below median results, we believe that there is a lot of room for improvement in our adhoc algorithm.

Run	Measure	Best	>= Median	< Median	Exact	Too Many	Too Few
att97fcuf1	Utility-1	7	37	10	0	20	27
att97feuf1	Utility-1	7	40	7	4	18	25
att97fcuf2	Utility-2	6	41	6	0	16	31
att97feuf2	Utility-2	10	41	6	2	10	35
att97fcasp	ASP	7	43	4	0	9	38
att97feasp	ASP	13	44	3	1	7	39

Table 12: Results for filtering runs

	D12345	DAT-1	DAT-2	DAT-3	DAT-4
Approximate Size (GB)	5.21	3.72	4.16	3.70	3.40
Indexing Time (Elapsed Minutes)	131	103	105	105	101
Index Size (GB)	1.81	1.21	0.88	1.10	1.20

Table 13: VLC Sub-collections

4 Filtering Runs

Our filtering track participation relies heavily upon our routing algorithm. Using the algorithm shown in Table 2 on the filtering track data, we learn a filtering query. Using this filtering query, we retrospectively rank the *training* collection and find a *similarity threshold* for the filtering query that would maximize our evaluation measure (utility or average set precision) on the *training* documents. Any test document that has a similarity greater than the above filtering threshold (to the filtering query) is assumed relevant and is passed to the user (if there were any). One should note that we optimize our filtering query to maximize average precision using DFO (step 4 in Table 2), and use the same query across evaluation measures. The only difference between different evaluation measures is in learning of the filtering threshold.

Table 12 shows the performance of our runs using the pooled evaluation. Runs att97fcuf1, att97fcuf2, and att97fcasp use the (conservative) one-pass algorithm from Table 2; whereas runs att97feuf1, att97feuf2, and att97feasp use the two pass algorithm (used in our routing run att97re). In general our filtering algorithm works well. The two-pass algorithm is somewhat better than our one-pass algorithm but we suspect that the difference is not statistically significant (we haven't done the tests yet!).

Also shown in Table 12 is an evaluation of our thresholding algorithm. The last three columns show how our threshold is doing as compared to an "ideal" threshold. The *Exact* column shows the number of queries for which our threshold did as well as the ideal threshold. The *Too Many* column shows the number of queries for which we retrieved more documents than we should have (so we had a lower threshold than the ideal threshold), and the last column shows the number of queries for which we had a higher threshold value than the optimal value. It is informative to know that the same thresholding algorithm does reasonably for utility-1 (3, -2, 0, 0), whereas for utility-2 (3, -1, -1, 0) and for average set precision, we seem to be retrieving too few documents in general. We plan to investigate our thresholding strategy in the near future, and possibly develop a more informed thresholding strategy.

5 VLC

To participate in the very large collection track, we have developed a new distributed version of the SMART retrieval system. The main design principle behind this version is: given a very large collection, it could be divided into several small, independent collections, which, when searched individually yield compatible document scores for a given query.

In the indexing phase, parts of the large collection are assigned to various CPUs (or machines on a LAN) as "independent" collections. The indexing is run in parallel on various CPUs. On our machine, the SMART system indexed the VLC text at about 2.4G/Hour. This could have been faster, had we limited ourselves

Average Query Length	27.48 words
Baseline Task P@20	0.348
Full Task P@20	0.530

Table 14: VLC Results

to running at most three indexing runs at a time instead of the five that we ran (since both the source text, and the indexed collection are stored on partitions of three striped disks and running more than three I/O bound processes usually slows down each of them due to disk bottleneck). We divided the VLC corpus into the following sub-collections: D12345, DAT-1, DAT-2, DAT-3, and DAT-4. (In retrospect, removing one of the disks from TREC D12345, and distributing it over DAT-1, DAT-3, and DAT-4 would have been a better distribution.) Table 13 shows some statistics for these sub-collection. Since our documents are *Lnu* weighted, we do not need term idf values at the time of indexing the collections, therefore all collections are indexed without any dependence on one-another. The total indexing time is the same as the longest time taken to index any sub-collection.

Once all the sub-collections are indexed individually, they read the dictionary and the df statistics for all other collections (df can possibly be encoded in the dictionary itself). Each collection merges the df information from all other collections to obtain a global df value (thus the idf value) for every term. Now, each collection has the true idf for every word. For the current implementation of the SMART system and for this task, this means reading about 50-75 MB of information from four other sources. Since all disks are local on our multi-processor system, this reading and merging took less than a minute for every collection. Of course, all this is possible since the stemming algorithm and the stop-word list is common across collections, the dictionaries across collections have same stems for a given word, and are therefore compatible.

For searching, a query is sent to each collection, and each collection retrieves its top twenty documents (twenty because this was the number wanted for evaluation in the VLC track). The similarities assigned to these documents are compatible since all collections have the global idf information for a term, as well as a common stemming/stopping algorithm (we are using *ltu* weighted queries, and we are using all sections—title, description, narrative—in the query, and we don't use phrases). The five lists of twenty documents each are merged, sorted by document score, and the top twenty documents are retrieved for evaluation. The whole retrieval take about two minutes for all fifty queries on our machine. The results are shown in Table 14. The full-task precision at twenty documents is very respectable, even better than the precision at twenty for the baseline task (a much smaller 2G database).

6 Conclusions

Our routing algorithm using query zones, word-pairs, and dynamic feedback optimization seems to be doing well. One big question that we should ask ourselves is why aren't we seeing much improvement in the routing performance over the last few TRECs? Doing a assessor agreement study in the routing environment would be interesting and might also tell us more about limits on our system performance.

Different components of our adhoc algorithm work well on different adhoc tasks. Overall, all components put together do yield noticeable improvements over a straight vector-match retrieval. As the adhoc task gets harder, with many queries with very few relevant documents, performance of the various components of our adhoc algorithm becomes unstable.

We show that it is feasible to index/retrieve-from very large text collections efficiently by sub-dividing them into smaller collections and sharing the collection information. We are encouraged by the retrieval effectiveness and the speed of our algorithms for very large collections.

Our routing algorithm followed by similarity thresholding seems to be doing a reasonable job of binary documents classification (filtering). Similarity thresholding should be studied more for the filtering environment.

Acknowledgments

We are thankful to David Lewis and Mandar Mitra for the useful discussions that helped us in various aspects of this work.

References

[1] Chris Buckley and Gerard Salton. Optimization of relevance feedback weights. In Edward Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 351–357. Association for Computing Machinery, New York, July 1995.

[2] Chris Buckley, Amit Singhal, and Mandar Mitra. Using query zoning and correlation within SMART: TREC-5. In D. K. Harman, editor, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, 1997 (to appear).

[3] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, 1996.

[4] D. K. Harman. Overview of the fourth Text REtrieval Conference (TREC-4). In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 1–24. NIST Special Publication 500-236, October 1996.

[5] Marti A. Hearst. Improving full-text precision on short queries using simple constraints. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages 217–232, Las Vegas, NV, April 1996.

[6] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In Hans-Peter Frei, Donna Harman, Peter Schauble, and Ross Wilkinson, editors, *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29. Association for Computing Machinery, New York, August 1996.

[7] Amit Singhal, Mandar Mitra, and Chris Buckley. Learning routing queries in a query zone. In Nick Belkin, Desai Narasimhalu, and Peter Willett, editors, *Proceedings of the Twentieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32. Association for Computing Machinery, New York, July 1997.

[8] E. M. Voorhees and D. K. Harman. Overview of the fifth Text REtrieval Conference (TREC-5). In D. K. Harman, editor, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, 1997 (to appear).

AT&T at TREC-6: SDR Track

Amit Singhal, John Choi, Donald Hindle, Fernando Pereira
AT&T Labs – Research
{singhal,choi,hindle,pereira}@research.att.com

Abstract

In the *spoken document retrieval* track, we study how higher word-recall—recognizing many of the spoken words—affects the retrieval effectiveness for speech documents, given that high word-recall comes at a cost of low word-precision—recognizing many words that were not actually spoken. We hypothesize that information retrieval algorithms would benefit from a higher word-recall and are robust against poor word-precision. Start-up difficulties with recognition for this task kept us from doing an systematic study of the effect of varying levels of word-recall and word-precision on retrieval effectiveness from speech. We simulated a high word-recall and a poor word-precision system by merging the output of several recognizers. Experiments suggest that having higher word-recall does improve the retrieval effectiveness from speech.

1 Introduction

From a retrieval system's perspective, a speech recognizer makes two types of recognition errors:

- **Omissions**: a spoken word is not recognized, and
- **Delusions**: the recognizer recognizes a word that was not spoken.

All recognition errors can be attributed to the above two types of errors, or their combination. *Omissions* reduce the word-recall, where word-recall is defined as the proportion of spoken words that are recognized; whereas *delusions* reduce the word-precision, where word-precision is defined as the proportion of recognized words that were spoken.

When speech-retrieval is done using word-based IR techniques, we hypothesize that omissions are much more hurtful than delusions. We believe that our IR techniques are quite robust against "noise" in the input text, given that there is enough "signal" in the text. High word-recall contributes to high signal in the text and high word-precision leads to low noise in the text. Therefore we want to study the effect of varying levels of word-recall and word-precision on retrieval effectiveness for speech.

Based on above hypothesis, we would like to enhance word-recall (by reducing omissions) at the cost of poorer word-precision. Two factors are responsible for omissions by a recognizer:

- Poor recognition: Often poor acoustics or language model constraints do not allow the recognizer to hypothesize a word with a reasonable confidence, even though the word is in the recognizer's vocabulary.
- Out of vocabulary (OOV): The spoken word is not in the recognizer's vocabulary, thus could never be recognized.

Using a word-based recognition system, we cannot attack the OOV problem, but we can certainly attack the other problem by generating many more words that are suggested by a recognizer even with a low confidence, and using these words for retrieval. As a recognizer suggests more and more words for a speech segment, the word-recall should improve but the word-precision should become poorer.

An attack on the OOV problem is to perform retrieval on sub-word acoustic units (phones, demi-syllables, syllables, or sequences of these units). [1, 5] For example, one might use all phone trigrams in the one-best phone transcription of the speech as the indexing units for an IR system. A user query could also be

translated into a bag of phone trigrams [1]. Given that even the best phone recognizers make a large number of mistakes, to improve phone trigram recall, we can once again use phone lattices to obtain the bag of phone trigrams for each speech document. Once the recognizer outputs a phone lattice, we can simply use all possible three-phone sequences in the lattice as indexing units. A similar lattice-based approach can be used for any class of indexing units, for example syllable or demi-syllable sequences.

2 Initial Plans

Since we did not already have a recognizer trained on HUB-4 material, we were relatively unconstrained with respect to recognizer design, so we set out to build a system that would attack both the word-recall and the OOV problems. We thus decided to implement a syllabic lattice recognition system, using existing training, recognition, syllabification and language-modeling programs. However, given that we started the work on our recognizer in late June, having a complete system running before the SDR deadline was quite an ambitious task.

For syllabic language modeling, we create a word list, generate word pronunciations with a text-to-speech system, and we apply a simple rule-based maximal onset syllabifier to the result to create a translation table from words to syllables. Since the position of a syllable within a word is quite informative for language modeling, we use four position-marked versions of each syllable: word-initial, word-medial, word-final and monosyllabic word. The resulting translation table from words to position-marked syllables is then used to translate the language-model training text into syllable sequences from which the appropriate n-gram statistics are computed.

To retrieve from syllabic-recognition output, the query words would be syllabified and the resulting syllable n-grams used to look up documents also indexed by syllable n-grams from built from the corresponding recognizer output. However, various difficulties described in the next section prevented us from having the full results of syllabic recognition in time for the track deadline, and we ended up using a simplified approach described later.

3 Recognizer

For recognition, we used phone-based models, a single-pronunciation dictionary, and a syllable bigram backoff language model.

For phone models, we used 3-state, left-to-right, HMMs with triphonic context dependence, trained on 39-dimensional acoustic feature vectors of mel-frequency cepstral coefficients and their first and second time derivatives centered on 5 and 3 frame windows, respectively. These vectors were initially modeled by a single full covariance Gaussian pdf (probability distribution function) per state, which was then rotated using the eigenvectors of the covariance matrix to remove correlations between parameters. Decorrelation was followed by the estimation of a weighted mixture of Gaussian pdfs with diagonal covariance. [3]

Context-dependency was modeled using categorical decision trees based on sub-phonemic classes, which effectively results in context-dependent tying of states. The decision trees were trained only on the training speech. A separate context-dependency model was defined for each training partition.

We built three separate sets of models: one set from speech labeled as high-fidelity with no background noise, one from medium and low fidelity speech with no background noise, and one from speech labeled as having background noise. In training each of the three sets of models, we bootstrapped from a single model trained on the channel-1 data from the NAB corpus.

For language modeling, we used a standard backoff bigram language model [2] over a vocabulary of about 20,000 position-marked syllables. This vocabulary size was chosen as a compromise between expected recognition speed and OOV rate. On the development test partition, a 20,000 word vocabulary yields an OOV rate 1.7%, while that for syllables is 0.4%. Position-marked syllables are represented by the their constituent phones together with a word boundary symbol, which is used in reconstructing words from the recognized

[1] For written queries, a text-to-speech system can be used to obtain the phone string corresponding to the query.

	IBM	AT&T+IBM
Word-Recall	69.3%	82.1%
Word-Precision	65.6%	18.9%

Table 1: Word-recall and word-precision for IBM's transcription and the merged transcription.

syllables. So, for example, the syllable that is typically spelled "bob" appears in the syllable wordlist as four distinct entries – `#_B_aa_B_#, #_B_aa_B, B_aa_B, B_aa_B_#` – corresponding to its appearance in the four words "Bob", "bobcat", "discombobulate", and "shishkabob", respectively.

The language model was trained on the SDR training corpus and the data from transcribed news broadcasts, designated for use in the baseline language model (LM) for the 1996 CSR Hub-4 evaluation. The syllable inventory was defined using all pronunciation alternates generated by our text-to-speech system. All syllables in the SDR training corpus were included in the syllable inventory, and all syllables with frequency greater than 3 in the Hub-4 LM corpus were included. To train the model, each word of the training text was mapped into its component syllables (including the word boundary symbols); for words with multiple pronunciations, a single alternate was randomly chosen for each occurrence.

4 Submitted Runs

Since this was our first experience with this particular material (AT&T had not participated in the HUB4 evaluations) and with a such a large material to be recognized, we encountered several difficulties that seriously curtailed our original experimental design.

First, we did not have at the time a reliable enough means of segmenting the test material into reasonably-sized segments of uniform type that could then be given to the appropriate one of our three recognizers (high quality, mid-low quality, and noisy). Therefore, we had to adopt the expedient of segmenting the test material into evenly-sized overlapping segments, and running all three recognizers on each segment. Second, the lattice recognizers that we had at the time were too slow to be able to recognize the whole test material in the available time and computing resources. Finally, the time and resources available to us were eroded further by a slew of unexpected systems problems.

Therefore, to submit a run we had to scale back our plans radically. Instead of lattice recognizers, we ran one-best recognizers (2-3 times real time) for the three models on all the test segments. Furthermore, even though we had all the machinery in place for extracting indexing units — syllable n-grams — from lattices, this machinery was not of any use for one-best transcriptions.

Both the "ad hoc" segmentation and the limited predictive power of the bi-syllable language model certainly contributed to the resulting poor recognition accuracy. While the segmentation into overlapping segments prevented us for computing word-error rates precisely, we estimated the word-error rate as high as 60%.[2]

Given all the problems we had with the recognizer, we had not time left to test our syllabic retrieval system. So we had to give up on attacking the OOV problem and revert back to using English words for retrieval. But since our recognition was syllabic, we had to translate all the "syllabic words" (mono-syllabic words and any syllable sequence that starts with a word-starting syllable, has any number of word-medial syllables, and ends in a word-ending syllable) into all possible English words using a pronunciation dictionary. This resulted in a mono-syllabic word `#_s_eh_n_t_#` generating the English words `cent, scent`, and `sent`. We applied this transformation to the recognizer output from each of our three acoustic models, resulting in three homophone-rich wordlists for every story. We then merged all three lists to get the final text to be indexed for a story, forming a coarse simile of a lattice.

The first run **att97sS1** was done using this merged list of words as a document and the user queries. To further simulate lattices, we created another set of words for every document by further merging the above merged list of words with the words that appeared in IBM's transcription of the speech. Our second retrieval run **att97sS2** was done using this longer list of words for a document, with higher word-recall and poorer

[2] In contrast, with a recently developed segmenter and a 20,000-word bigram model, the word-error rate went down to 40% even without changing the acoustic models.

word-precision. Table 1 shows the recall and the precision figures for the baseline (IBM's) transcription, and the merged (AT&T+IBM) transcription used in att97sS2. These figures were computed using non-stop words (because only they matter in retrieval), and by ignoring word frequency (since we use binary tf weighting). The word-recall and the word-precision was computed for every story and was further averaged across stories. We observe that the merged list does exhibit a higher word-recall and has a much poorer word-precision than IBM's transcription. Our main motivation for doing this merging was that if the merged retrieval run works better than both att97sS1, and att97sB1 (which is a retrieval run done solely on IBM's baseline word transcriptions), then our hypothesis that improving word-recall should help speech retrieval effectiveness will be supported.

We use an internally modified version of Cornell's SMART system for retrieval. We used standard inner-product similarity to rank the *bnu* weighted documents using *ltu* weighted queries within the SMART system. [4] Where the weight of a word in a document (bnu) is:

$$\frac{1}{0.8 + 0.2 \times \frac{number\ of\ unique\ words\ in\ document}{average\ number\ of\ unique\ words\ per\ document}}$$

and the weight of a query word is (ltu):

$$\frac{1 + log(tf) \times log(\frac{N+1}{df})}{0.8 + 0.2 \times \frac{number\ of\ unique\ words\ in\ query}{average\ number\ of\ unique\ words\ per\ document}}$$

5 Results

Out of the three evaluation measures being used for known-item searching — mean rank, mean reciprocal rank, and counts of how many known items were found within top 5, 10, 20 and 100 documents — the first two (mean rank and mean reciprocal rank) have problems in our view. Mean rank is heavily influenced by even a single miss (very poorly ranked document, an outlier). For example, if the known item for a query is ranked 200, the mean rank for the entire collection of 49 queries drops by almost 4, irrespective of how well the system is retrieving for the other 48 queries. However, if outliers are removed, *i.e.*, all queries for which a system has extremely poor results (under some definition of extremely poor), then average rank might yield meaningful results. Mean reciprocal rank, on the other hand, differentiates too much between a known-item being ranked at rank 1 vs. if the known-item is ranked at rank 2. From a user's perspective, we believe that ranking the known-item at rank 1 is not 100% better than ranking it at rank 2, a ratio assigned by mean reciprocal rank. We believe that counts of how many known items were found within top 5, 10, 20 and 100 documents is the most meaningful measure out of the above three evaluation measures. If one has to compare only two runs, another meaningful comparison would be a query-by-query comparison of the two systems on a scatter plot. This would enable us to view which system is performing better on most of the queries and by how much.

Figure 1 shows a histogram of how document are ranked when different texts — the human transcription (Human), IBM's transcription (IBM), our merged list of words (AT&T), and our list merged with IBM's words (AT&T+IBM) — are used in retrieval for the 49 user queries. Our first observation from Figure 1 is that retrieval done over the output of a speech recognizer using conventional IR techniques is quite respectable. This agrees with the observation of other researchers who have worked with other speech corpora. As expected, our internal recognition does not perform as well as the other transcriptions. We are actually surprised that it works as well as it does. Given the recognition difficulties described above, it is somewhat surprising that our system still retrieves thirty three answer documents within top five using our merged list of words, suggesting that the task at hand was rather easy.

More interestingly, we observe that once we merge the baseline transcription provided by IBM and our list of words, even though we retrieve the answer document in the top five documents for fewer queries (which we believe is a reflection upon the poor quality of our recognition), if we look in the top ten documents, retrieval from the merged transcription (AT&T+IBM) outperforms retrieval from IBM's transcription alone. This is true even when we look in the top twenty documents. Actually, when looked in the top twenty documents, the merged transcription works as well as the human transcription. Forty six out of forty nine queries have their

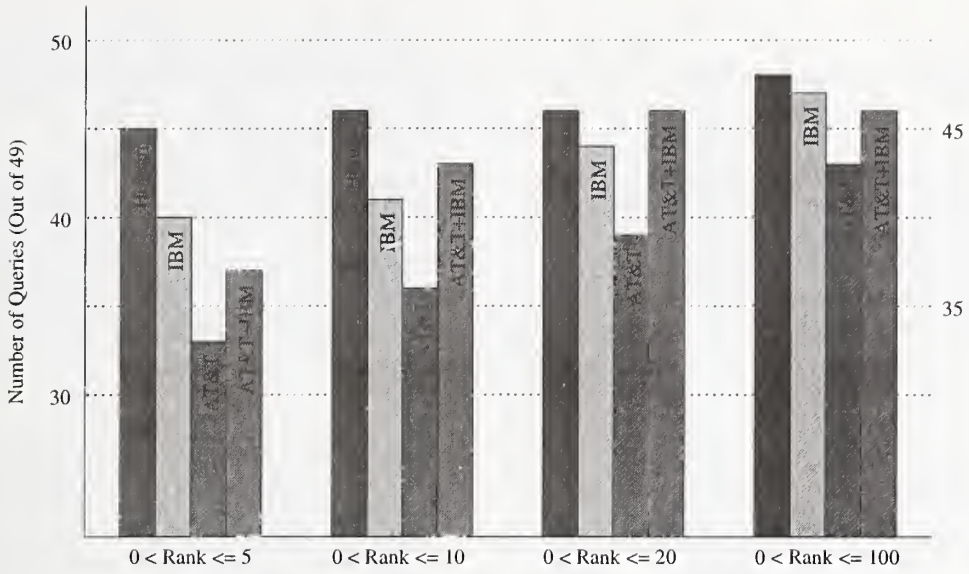


Figure 1: Comparison of retrieval from various transcriptions.

answers listed in the top twenty for retrieval from both the human and the merged transcription. Of course, where the answer is within the top twenty is also important. We believe that if we had a better transcription internally, these results could have been better. This results are encouraging for further experimentation using word and sub-word lattices.

If we remove the outlier queries, *i.e.*, query 3 (for which the answer article is ranked at ranks 236, 389, and 178 for the human, IBM, and AT&T+IBM transcriptions, respectively), query 23 (known item rank is 222 for AT&T+IBM), and query 42 (IBM's transcription does not retrieve the answer at all), then the average rank of the known item for the human, IBM, and AT&T+IBM transcription are 2.65, 4.15, and 3.04, respectively. This once again indicates that retrieval from AT&T+IBM transcription is somewhat better than retrieval from IBM's transcription alone. This lends further support to possibility of improved retrieval using lattices.

Another evidence that retrieval from AT&T+IBM transcription is better than the retrieval from the IBM's transcription alone is shown in Figure 2. Figure 2 shows what the rank of an answer document is using the AT&T+IBM transcription vs. the rank of the corresponding document using IBM's transcription alone. The x-axis is the rank of the answer document as retrieved from IBM's transcription (log-scale), and the y-axis is the rank of the answer document as retrieved from AT&T+IBM transcription (log-scale). A point below the diagonal line indicates that the rank of the answer document was lower (better retrieval) for the AT&T+IBM transcription. This scatter plot shows that, in general, the merged transcription has better results. 24 of the 49 queries have their known-item retrieved at identical ranks for the two system. For 16 queries, retrieval from AT&T+IBM transcription is better, and for 9 queries retrieval from AT&T+IBM transcription is worse than retrieval from IBM's transcription alone.

6 Directions

We have recently finished implementing a fast lattice recognizer, and are currently in the process of training new acoustic models. We have also developed a speech segmenter internally that assigns portions of the test speech to one of several possible acoustic categories in our system. We plan to investigate lattice based recognition in a much more organized manner in the near future.

Even though known-item retrieval is a fine task for initial evaluation of speech retrieval system, the small size of speech corpora (as compared to more traditional information retrieval corpora) makes this task artificially easy. There is very little noise in the corpora. Any user query hits just a few documents, if at

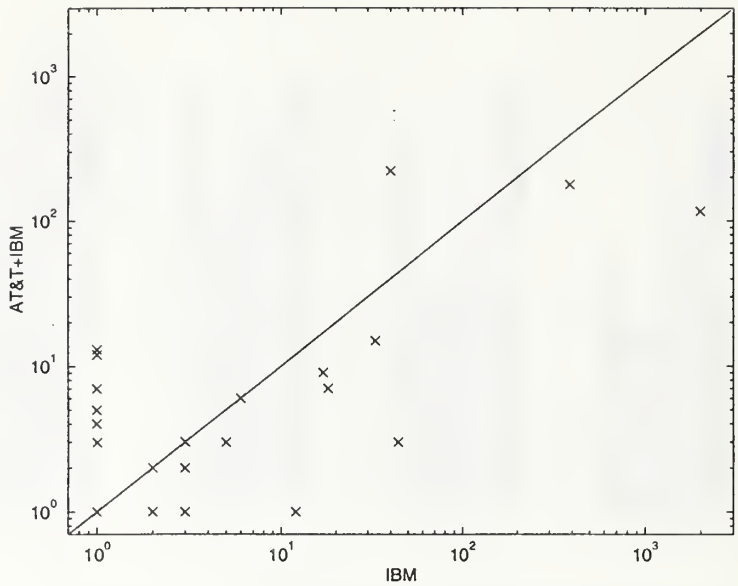


Figure 2: Comparison of retrieval from IBM's transcription and AT&T+IBM transcription.

all it hits any. Therefore, larger speech databases are always desirable in a speech retrieval task. Moving to a more traditional, ranking evaluation using average precision might also exemplify some strengths and shortcomings of various approaches of speech retrieval.

Acknowledgments

We are very grateful to Andrej Ljolje, Mehryar Mohri, and Michael Riley for all their help in building the recognizer for this data.

References

[1] G.J.F. Jones, J.T. Foote, K. Sparck Jones, and S.J. Young. Retrieving spoken documents by combining multiple index sources. In Hans-Peter Frei, Donna Harman, Peter Schauble, and Ross Wilkinson, editors, *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 30–38. Association for Computing Machinery, New York, August 1996.

[2] S.M. Katz. Estimation of probabilities from sparse data from the language model component of a speech recognizer. *IEEE Transactions of Acoustics, Speech and Signal Processing*, pages 400–401, 1987.

[3] Andrej Ljolje. The importance of cepstral parameter correlations in speech recognition. *Computer Speech and Language*, 8, 1994.

[4] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In Hans-Peter Frei, Donna Harman, Peter Schauble, and Ross Wilkinson, editors, *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29. Association for Computing Machinery, New York, August 1996.

[5] M. Wechsler and P. Schauble. Indexing methods for a speech retrieval system. In C.J. van Rijsbergen, editor, *Proceedings of the MIRO Workshop*, 1995.

Automatic 3-Language Cross-Language Information Retrieval with Latent Semantic Indexing

Bob Rehder
Dept. of Psychology
Inst. of Cognitive Science
U. of Colorado, Boulder
Boulder, CO 80309
rehder@psych.colorado.edu

Michael L. Littman
Dept. of Computer Sci.
Duke University
Durham, NC 27708
mlittman@cs.duke.edu

Susan Dumais
Microsoft Research
One Microsoft Way
Redmond WA, 98052
sdumais@microsoft.com

Thomas K. Landauer
Dept. of Psychology
Inst. of Cognitive Science
U. of Colorado, Boulder
Boulder, CO 80309
landauer@psych.colorado.edu

Abstract

This paper describes cross-language information-retrieval experiments carried out for TREC-6. Our retrieval method, cross-language latent semantic indexing (CL-LSI), is completely automatic and we were able to use it to create a 3-way English-French-German IR system. This study extends our previous work in terms of the large size of training and testing corpora, the use of low-quality training data, the evaluation using relevance judgments, and the number of languages analyzed.

Introduction

Cross-language LSI (CL-LSI) is a fully automatic method for cross-language document retrieval in which no query translation is required. Queries in one language can retrieve documents in other languages (as well as the original language). This is accomplished by a method that automatically constructs a multi-lingual semantic space using latent semantic indexing (LSI); this semantic space is exploited in the form of a *vector lexicon*, which assigns each word in each language to a point in the high-dimensional space.

For the CL-LSI method to be used, an initial sample of documents in one language must be available with "mates" in all other languages. In past work, these mates were created by human translators; in the present work, we used a combination of machine translation and automatic mate selection from a comparable corpus to create mate sets. An LSI analysis of the set of documents and mates results in a multi-language *semantic space* in which terms from all languages are represented. Concretely, this semantic space takes the form of a vector lexicon in which each word in each of the languages is assigned a high-dimensional vector representation. Queries in any language can retrieve documents in any language without the need to translate the query because all text records (documents and queries) are represented as language-independent numerical vectors in the same semantic space.

The present work builds on our past experience with CL-LSI by

1. scaling to larger document collections than had been previously attempted,
2. using much noiser training data (no human translations) than had been previously attempted, and
3. using more languages than had been previously attempted (3 instead of 2).

We explored a completely automatic approach to information retrieval between topics in English, French, and German and documents in English, French, and German. To train our system, we began with a coarsely parallel aligned collection of over 80,000 German and French documents provided by NIST, which we used to train an initial German-French cross-language retrieval system. The German-French training pairs that were assigned the lowest similarity scores by our initial system were discarded in an attempt to weed out document pairs that were not properly aligned. The remaining 40,000 French-German pairs were then augmented with computer-generated English translations of the German documents (also provided by NIST), to create a collection of 40,000 3-language mate triplets, which we used to train an English-French-German retrieval system. Thus, in contrast to most previous work on automatic cross-language IR, no human translations were used in training. We used our retrieval system to compare both short and long queries in all three languages against the full set of English, French, and German documents. We feel that, by analogy to our experience with monolingual LSI, our approach is likely to show the largest benefits for the short topics, but this has been difficult to assess so far. The noteworthy aspects of our approach are that it is completely automatic, it works between any pair of the three target languages, it is trained using an imperfectly aligned collection, and it exploited a simple "bootstripping" technique to help make the most of noisy training materials.

Background

This section provides background on latent semantic indexing (LSI) and its cross-language extension. Other introductions are also available (Deer-

wester *et al.* 1990; Berry, Dumais, & O'Brien 1995; Dumais 1995).

Latent Semantic Indexing Motivation

Latent semantic indexing is a variant of the vector-space method (Salton & McGill 1983) in which the dependencies between terms are explicitly modeled and exploited to improve retrieval. One advantage of the LSI representation is that a query can retrieve a relevant document even if they have no words in common.

Most information-retrieval methods depend on exact matches between words in users' queries and words in documents. Typically, documents containing one or more query words are returned to the user. Such methods will, however, fail to retrieve relevant materials that do not share words with users' queries. One reason for this is that the standard retrieval models (e.g., Boolean, standard vector, probabilistic) treat words as if they are independent, although it is quite obvious that they are not. A central theme of LSI is that term-term inter-relationships can be automatically modeled and used to improve retrieval; this is critical in cross-language retrieval since direct term matching is of little use.

LSI examines the similarity of the "contexts" in which words appear, and creates a reduced-dimension feature-space representation in which words that occur in similar contexts are near each other. That is, the method first creates a representation that captures the similarity of usage (meaning) of terms and then uses this representation for retrieval. The derived feature space reflects these inter-relationships. LSI uses a method from linear algebra, singular value decomposition (SVD), to discover the important associative relationships. It is not necessary to use any external dictionaries, thesauri, or knowledge bases to determine these word associations because they are derived from a numerical analysis of existing texts. The learned associations are specific to the domain of interest, and are derived completely automatically.

The singular-value decomposition (SVD) technique is closely related to eigenvector decomposition and factor analysis (Cullum & Willoughby 1985). For information retrieval and filtering applications we begin with a large term-document matrix, in much the same way as vector-space or Boolean methods do. This term-document matrix is decomposed into a set of k, typically 200–300 in monolingual applications, orthogonal factors from which the original matrix can be approximated by linear combination; this analysis reveals the "latent" structure in the matrix that is obscured by noise or by variability in word usage.

The result of the SVD is a set of vectors representing the location of each term and document in the reduced k-dimension LSI representation. Retrieval proceeds by using the terms in a query to identify a point in the space—technically, the query is located at the weighted vector sum of its constituent terms. Documents are then ranked by their similarity to the query, typically using a cosine measure of similarity. While the most common retrieval scenario involves returning documents in response to a user query, the LSI representation allows for much more flexible retrieval scenarios. Since both term and document vectors are represented in the same space, similarities between any combination of terms and documents can be easily obtained—one can, for example, ask to see a term's nearest documents, a term's nearest terms, a document's nearest terms, or a document's nearest documents. We have found all of these combinations to be useful at one time or another.

In monolingual document-retrieval tests, the LSI method has equaled or outperformed standard vector methods in almost every case, and was as much as 30% better in some cases (Deerwester *et al.* 1990; Dumais 1995).

Latent Semantic Indexing Mathematics

LSI begins with a collection of m documents containing n unique terms and forms an $n \times m$ sparse matrix E, with E_{ij} containing a value related to the number of times term i appears in document j. Various weighting schemes can be applied to the raw occurrence counts; in this work, we used log-entropy weighting ($\log(\mathrm{tf}+1)\mathrm{entropy}$).

Once the document-term matrix E has been created, LSI computes the similarity between two text objects (a query and a document, say) as follows. First, a text object q is represented by an $n \times 1$ vector, much like a column of the E matrix and with the same sorts of term weighting applied. Next, the similarity between text objects q_1 and q_2 can be computed, typically by cosine scoring; in the vector-space method, this can be represented as $\mathrm{sim}(q_1, q_2) = q_1^T q_2 / \sqrt{q_1^T q_1 \cdot q_2^T q_2}$.

A mathematically useful way of viewing the process of computing text-object similarity scores in the vector-space method is this. Each of the n terms in the collection has a vector representation, specifically term i is an $n \times 1$ vector of zeros with a 1 in component i. The representation of a text object q is a weighted sum of the term vectors of the terms that appear in the text object. Thus, the similarity between text objects q_1 and q_2 is

$$\mathrm{sim}(I_n q_1, I_n q_2), \tag{1}$$

where I_n is the $n \times n$ identity matrix. Here, I_n plays the role of a *vector lexicon*, in that it assigns each term a vector "definition." Of course, pre-multiplying by the identity matrix in Equation 1 does not change the comparison in any way; by

using other vector lexicons, we can substantially change the way similarities are computed. Note that the only role played by the document-term training matrix E in the vector-space method is in the computation of weighting factors for the components of text objects.

LSI can be viewed very similarly to the vector-space method. LSI also begins with the formation of the term-document matrix E. Then, the E matrix is analyzed using singular value decomposition (SVD) to extract structure concerning document-document and term-term correlations. Mathematically, an SVD of E can be written

$$E = U(E)\ \Sigma(E)\ V(E)^T, \tag{2}$$

where $U(E)$ is an $n \times n$ matrix such that $U(E)^T U(E) = I_n$, $\Sigma(E)$ is an $n \times n$ diagonal matrix of *singular values* and $V(E)$ is an $n \times m$ matrix such that $V(E)^T V(E) = I_m$. This assumes for simplicity of exposition that E has fewer terms than documents, $n < m$.

This SVD analysis can be used to construct lower rank approximations of E, and this is how it is typically used in the context of LSI. Reducing the rank of the approximation results in a synonym-collapsing effect in practice. It also reduces the total amount of processing and storage associated with preprocessing and retrieval. We write

$$E_k = U_k(E)\ \Sigma_k(E)\ V_k(E)^T, \tag{3}$$

to denote the components of the k-dimensional SVD and its rank-k reconstruction of E.

The $U_k(E)$ matrix in Equation 3 can be used as an alternative vector lexicon to the I_n in Equation 1 in that it assigns a vector representation to every term in the term-document matrix E. Thus, in LSI, the k-dimensional similarity between text object q_1 and text object q_2 in the context of E is

$$\mathrm{sim}(U_k(E)^T q_1, U_k(E)^T q_2). \tag{4}$$

Berry, Dumais, & O'Brien (1995) give justifications for the use of the matrix of left singular vectors $U_k(E)$ as a vector lexicon.

Cross-language LSI

The techniques of mono-lingual LSI transfer easily to the cross-language case simply by using a different notion of the term-document matrix (Landauer & Littman 1990).

For concreteness, let E be a term-document matrix of m English documents and n^E English terms, F be a term-document matrix of m semantically equivalent French documents and n^F French terms, and G be a term-document matrix of m semantically equivalent German documents and n^G French terms. These documents are mate-aligned, in the sense that document $1 \leq i \leq m$ in the English collection is directly related to document i in the French and German collections. The multi-language term-document matrix

$$M = \begin{bmatrix} E \\ F \\ G \end{bmatrix}$$

is an $(n^E + n^F + n^G) \times m$ matrix in which column i is a vector representing the English, French, and German terms appearing in the union of document i expressed in all three languages.

Cross-language LSI (CL-LSI) begins with the matrix M and performs an SVD,

$$M = \begin{bmatrix} U_k^E(M) \\ U_k^F(M) \\ U_k^G(M) \end{bmatrix}\ \Sigma_k(M)\ V_k(M),$$

where $U_k^E(M)$, $U_k^F(M)$, $U_k^G(M)$ are k-dimensional vector lexicons for English, French, and German, respectively. Empirically, similar English, French, and German words are given similar definitions, so this vector lexicon can be used for cross-language retrieval. In particular, consider an English text object q_E and a French text object q_F. They can be compared using the obvious generalization of Equation 4,

$$\mathrm{sim}(U_k^E(M)^T q_E, U_k^F(M)^T q_F). \tag{5}$$

Note that, in our experiments, we chose not to take advantage of cross-language homonyms. That is, the word "documents" in French was treated distinctly from the word "documents" in English. The same holds true of names and numbers. For this collection of languages, it is likely that identifying and exploiting cross-language homonyms would improve performance. We chose not to do this so that we could better evaluate how well CL-LSI was able to identify patterns in word usage between the languages without relying on cognates or other "incidental" properties of the languages used in this study.

Previous Evaluations of CL-LSI

In past work, we have used a number of informal evaluation techniques to help determine whether retrieval systems created using CL-LSI can effectively compare text objects between languages. In the *overlap* technique (Landauer & Littman 1990), we began with a set of several thousand English and French mates and compared each with a set of English queries to determine the 10 best matching French documents and 10 best matching English documents to each query. We then counted, for each query, the number of mates in common in the English and French return sets, and found that an average of 4.1 mate pairs appeared in the return sets. Thus, to the extent that CL-LSI is able to match English queries to English documents, it is

also able to match English queries to French documents nearly as well.

In *mate-retrieval* evaluation, we again begin with a test set of English and French mates. Next, we take each English document as a query and compute the rank of its French mate when the English "query" is compared with each French document. While this use of long queries does not provide a very accurate measure of the performance of CL-LSI in a real retrieval setting, it does give some indication as to whether the language-independent vector representation of meaning is at all reasonable. A typical result (Dumais, Landauer, & Littman 1996) is that CL-LSI returns cross-language mates over 98% of the time from a test set of 1,500 mates. Less strong, but still impressive, results are obtained using imperfectly matched or machine translated mates for training, and mismatches between training and testing data (Dumais, Littman, & Landauer 1997 to appear).

CL-LSI has been evaluated in a more traditional *relevance-judgment* experiment (Carbonell *et al.* 1997). The implementation of CL-LSI in that study was compared to the generalized vector-space method, example-based query translation, and pseudo-relevance feedback query expansion, as well as a number of other techniques, and fared relatively poorly. The version of CL-LSI in our work differs from that in the Carbonell *et al.* (1997) study in our use of Equation 5 (instead of $\mathrm{sim}(U_k^E(M)^T\Sigma_k(M)^{-1}q_E, U_k^F(M)^T\Sigma_k(M)^{-1}q_F)$) and in the number of dimensions used (we tend to use 500-1500 dimensions for cross-language comparisons, they used 200).

In our recent studies with the Carbonell *et al.* (1997) collection, CL-LSI outperformas all methods except example-based query translation.

CL-LSI in TREC-6

The document collection in the TREC-6 cross-language track experiments consisted of English, German, and French newspaper articles from 1988-1990. Specifically:

- English (242,918 documents, 684MB): Associated Press newswire (AP)
- German (185,099 documents; 269MB): Schweizerischen Depeschenagentur, Swiss news agency (SDA-G)
- German (66,741 documents, 176MB): Neue Zuercher Zeitung, Swiss German newspaper (NZZ)
- French (141,656 documents; 199MB): Schweizerischen Depeschenagentur, Swiss news agency (SDA-F).

A total of 25 topics were created by NIST in each of English, French, and German. We used an initial alignment, bootstrip cleaning, and machine translation to create an English-French-German CL-LSI system. Each of these steps are explained in more detail in the following sections.

Initial Alignment

For CL-LSI to apply in English, French, and German, it is necessary to have a set of documents in one language with mates in both of the others. In previous studies, corpora were used in which human translators had generated these mates. In this study, we used automatic methods to generate these mates. Note that neither the alignment (Sheridan & Ballerini 1996), nor the machine translation was actually carried out by our group. We simply used the results of the work of others.

To begin, the SDA-G set (185k German documents) was taken as the core set of training documents. For each of these, the SDA-F set (142k French documents) was searched for possible mates. As truly accurate mate decisions would require human judgment, a simple rule of thumb was used. A French document was declared as a mate for a German document if the two documents (newswire articles) appeared on the same day and had a sufficient number of words in common in their keyword fields. Any German document with no mates found through this procedure was removed from the core set of training documents. The result was a set of 83,698 German documents and their automatically discovered French mates.

Note that it is possible for a single French document to occur more than once as a mate if it happened to align well with more than one German document. It is estimated that 20% of the aligned German documents are paired with a non-unique French document (a French document that is aligned with some other German one). Also, in our preliminary look at these alignments, we coarsely estimate that 10% of the 84k document pairs are misaligned. For example, one pair consists of articles about a bus bombing and a revenge killing in Jerusalem on the same day. These stories are not really related, but this is impossible to tell based on the keywords (terrorism, Jerusalem), or the date.

Bootstrip Cleaning

As an attempt to automatically weed out this sort of error, we carried out the following "bootstripping" procedure.

1. Separate the 83,698 French-German document pairs into a verification set of 3k documents and a training set of 80,698 documents.
2. Use CL-LSI on the 80,698 training documents to create a vector lexicon for German and French. As a check, calculate mate-retrieval scores for the

Training-set size	~80k	40k	20k
Dimensions	500	1000	1200
Average cosine of mates	0.508	0.469	0.459
Average rank of mate	3.00	2.51	2.73
Average log rank	0.287	0.274	0.325
Proportion in top one	80.1%	82.5%	79.8%
Proportion in top ten	96.7%	96.6%	95.6%
Proportion in top fifty	99.6%	99.4%	99.4%
Number not in top 5%	4	1	1

Table 1: Mate-retrieval results for bootstripping experiment

Training set	F-G	E-F-G	E-F-G
Testing set	G-F	G-F	E-F
Dimensions	1000	800	800
Average rank of mate	2.51	2.76	2.97
Average log rank	0.274	0.308	0.284
Proportion in top one	82.5%	80.5%	82.2%
Proportion in top ten	96.6%	96.1%	96.0%
Proportion in top fifty	99.4%	99.4%	99.5%
Number not in top 5%	1	2	4

Table 2: Three-language mate-retrieval scores

3k verification documents using the vector lexicon (see Table 1).

3. Calculate similarity scores for each of the 80,698 training documents using the derived vector lexicon. Identify the 40k pairs with the highest similarity scores.

4. Use CL-LSI on the best 40k training documents to create a new vector lexicon for German and French. As a check, calculate mate-retrieval scores for the 3k verification documents using the new vector lexicon.

5. Calculate similarity scores for each of the 40k training documents using the derived vector lexicon. Identify the 20k pairs with the highest similarity scores.

6. Use CL-LSI on the best 20k training documents to create a new vector lexicon for German and French. As a check, calculate mate-retrieval scores for the 3k verification documents using the new vector lexicon.

Looking at the average rank of mate reported in Table 1, we see that performance improves in going from 80k training documents to 40k, then degrades at 20k. Most of the other scores follow a similar trend. Next, we give a brief explanation of the measures given in Table 1. The average pairwise cosine gives the similarity between the mates in the 3k-document verification set. Log ranks were computed simply to diminish the weight given to pairs with very poor rankings (surpress the effect of outliers).

The best of these three runs appears to be the one with the 40k-document training set. It is this set we use in the remainder of our experiments. Mate-retrieval performance is quite strong for this training set: 82% of the mates have the highest cosine, 97% of mates are in the top 10, and over 99% are in the top 50. While there are obviously some pairs that are assigned poor similarity scores, there is only 1 document pair (out of 3k) that had a rank not in the top 5%, which is reasonable.

Machine Translation: Extending to 3 Languages

After the bootstrip cleaning, we were left with an French-German retrieval system. To extend this to include English, we made use of machine-translations of the SDA-G documents. Of the 40k German documents in our core training set, 12 of them did not have available translations into English. Therefore, we carried out a CL-LSI analysis of a 39,988-document collection of German, with a French and English mate for each German document. (Note that preliminary experiments (Littman & Keim 1997) indicate that it is important for 3-language training collections to use complete sets of mates.)

Table 2 gives mate-retrieval results for the resulting 800-dimensional three-way retrieval system. The German-to-French mate-retrieval performance degrades slightly for the 3-way system compared to the French-German system from the bootstapping experiment; it is not clear whether this is due to the inclusion of English or the decreased number of dimensions. Nevertheless, the 3-way system exhibits respectable English-to-French mate-retrieval performance—comparable to the German-to-French performance. This is remarkable, given that the statistical relationship between English and French in this system is very indirect: French document i was paired with English document i only because of the coarse alignment between German i and French i and the fact that German i was machine translated to yield English i.

Results

A noteworthy feature of the CL-LSI approach is that, because all words from all languages exist together in a common high-dimensional space, queries can be matched against documents in all languages together; without specifying the target language, a user receives first a document in whichever language gives the best match. While this feature was not exploited in the TREC studies, here is a simple illustration.

Figure 1 gives the English version of one of the topics. We issued a long query (the union of the

short: Reasons for controversy surrounding Waldheim's World War II actions.
long: Revelations about Austrian President Kurt Waldheim's participation in Nazi crimes during World War II are argued on both sides. Relevant documents are those that express doubts about the truth of these revelations. Documents that just discuss the affair are not relevant.

Figure 1: Long and short forms of topic 1

long and short version of the topic) against the entire test collection (all three languages). The top ten documents (and the 50th) retrieved, along with their language and similiarity score, are:

1. (German: 0.538): Waldheim als "Mitwisser," nicht als "Mitschuldiger."
2. (English: 0.532) Waldheim Says Pope Visit Will Help Austria
3. (English: 0.531) Former Chancellor To Be Charged With Perjury

   Former Chancellor Fred Sinowatz will be charged with perjury in connection with testimony in a 1987 trial that arose from a probe into President Kurt Waldheim's World War II past, the justice minister said Wednesday.
4. (English: 0.521) Austria Marks Annexation by Nazi Germany
5. (English: 0.515) Document Shows Waldheim Knew of Plan To Deport Greeks To Labor Camps
6. (English: 0.515) Documents Show Waldheim Transcribed, Forwarded Order To Kill Partisans
7. (German: 0.509) Waldheim erinnert an das "tragische Ereignis" des Jahres 1938.
8. (German: 0.507) Waldheim: Ungenaue Angaben zu Nazi-Vergangenheit gang und gäbe.
9. (English: 0.505): Austrian President Withdraws Lawsuit Against WJC President Bronfman
10. (English: 0.504) World Jewish Congress Calls Austrian Reparations To Jews "Desecration"

...

50. (French: 0.444) Publication d'un dossier sur le pass nazi de Kurt Waldheim.

We see that three German documents and seven English (AP) documents make up the top ten, all from 1988, as it turns out. The titles (or title and first paragraph) of all the stories indicate that they have to do with Waldheim. Recall that this is acheived without ever explictly matching on "Waldheim"—the term was considered separately in each of the three languages.

RUN TAG	TOPIC	LANG	TARGET
97lsiLGG	long	G	G
97lsiLFF	long	F	F
97lsiLEE	long	E	E
97lsiSGG	short	G	G
97lsiSFF	short	F	F
97lsiSEE	short	E	E

Table 3: List of monolingual runs produced

RUN TAG	TOPIC	LANG	TARGET
97lsiLGF	long	G	F
97lsiSGF	short	G	F
97lsiLFG	long	F	G
97lsiSFG	short	F	G
97lsiLEG	long	E	G
97lsiSEG	short	E	G
97lsiLEF	long	E	F
97lsiSEF	short	E	F
97lsiLGE	long	G	E
97lsiSGE	short	G	E
97lsiLFE	long	F	E
97lsiSFE	short	F	E

Table 4: List of single-language cross-language runs produced

In the returned list, we can see that the English query brought back predominantly English documents. Nonetheless, the best-matching document in the set is in German.

Catalog of Runs

For our submitted runs, we issued both long and short queries in each of the three languages against the test documents in each of three languages and returned the top 1000 for each. The actual runs are listed in Tables 3 and 4.

Preliminary results of relevance judgments are given in Table 5. The table lists, for each of the topics, the fraction of runs we submitted for which the average precision was at or above median compared to those submitted by other groups. Topics are listed in decreasing order of performance for CL-LSI.

These early results look particularly bad, especially in the monolingual case. This level of performance is far below LSI's typical performance on monolingual tasks, so we are looking for an explanation of this. It is interesting to note, however, that despite CL-LSI's overall poor early showing, it does relatively better on cross-language runs than on monolingual runs. We look forward to getting a more complete set of relevance judgments and to spending more time understanding the situations in which CL-LSI had difficulty identifying relevant documents.

Topic	Mono	Cross	Avg.
10	.33	.67	.56
6	.00	.83	.55
1	.67	.42	.50
19	.17	.58	.44
9	.00	.58	.39
14	.00	.42	.28
17	.00	.17	.11
18	.00	.17	.11
24	.00	.17	.11
5	.00	.17	.11
11	.00	.08	.05
16	.00	.08	.05
2	.00	.08	.05
average	.09	.34	.25

Table 5: Average number of runs at or above median (by topic)

References

Berry, M. W.; Dumais, S. T.; and O'Brien, G. W. 1995. Using linear algebra for intelligent information retrieval. *SIAM Review* 37(4):573–595.

Carbonell, J.; Yang, Y.; Frederking, R.; Brown, R. D.; Geng, Y.; and Lee, D. 1997. Translingual information retrieval: A comparative evaluation. In *Proceedings of Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97).*

Cullum, J. K., and Willoughby, R. A. 1985. Chapter 5: Real rectangular matrices. In *Lancozos algorithms for large symmetric eigenvalue computations - Vol 1 Theory.* Boston: Brikhauser.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. A. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Dumais, S. T.; Landauer, T. K.; and Littman, M. L. 1996. Automatic cross-linguistic information retrieval using latent semantic indexing. SIGIR96 Workshop On Cross-Linguistic Information Retrieval.

Dumais, S. T.; Littman, M. L.; and Landauer, T. K. 1997, to appear. Automatic cross-language information retrieval using latent semantic indexing. In Grefenstette, G., ed., *Cross Language Information Retrieval.*

Dumais, S. T. 1995. Using LSI for information filtering: TREC-3 experiments. In Harman, D., ed., *The Third Text Retrieval Conference (TREC3)*, 219–230. National Institute of Standards and Technology Special Publication 500-225.

Landauer, T. K., and Littman, M. L. 1990. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research.* 31–38.

Littman, M. L., and Keim, G. A. 1997. Cross-language text retrieval with three languages. Technical Report CS-1997-16, Department of Computer Science, Duke University.

Salton, G., and McGill, M. J. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill.

Sheridan, P., and Ballerini, J. P. 1996. Experiments in multilingual information retrieval using the spider system. In Frei, H.-P.; Harman, D.; Schäble, P.; and Wilkinson, R., eds., *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96*, 58–65.

MDS TREC6 Report

Michael Fuller Martin Kaszkiel Chien Leng Ng Phil Vines Ross Wilkinson
Justin Zobel

Department of Computer Science
RMIT, GPO Box 2476V
Melbourne VIC3001, Australia
msf,cln,martin,vines,ross,jz@mds.rmit.edu.au

1 Introduction

This year the MDS group has participated in the ad hoc task, the Chinese task, the speech track, and the interactive track. It is our first year of participation in the speech and interactive tracks. We found the participation in both of these tracks of great benefit and interest.

2 Full Description of Techniques

In this section of the paper we will give as complete a description as we can of our methodology. We do so by describing the following: term definition, casefolding, stopping, and stemming. This defines the terms that we use. We then give the formula used for matching. After this we give exact descriptions of how we carry out passage retrieval, term expansion, and combination.

A term is a sequence of characters chosen from the alphabet {a–z,A–Z,0–9}. The sequence has a maximum length of 256 but if the string consists solely of numbers a maximum length of 4 applies. All other characters are treated as term delimiters.

To casefold, all uppercase letters are converted to their lowercase equivalents.

To stop, we remove all terms that are in the list given in the appendix.

Terms are stemmed as given in the Lovin's algorithm[4].

Passages are formed by sequences of words that are 50, 100, 150, 200, 250, 300, 350, 400, 450, 500,

550 or 600 words long. Passages may commence at any 25 word interval. Every such passage is then treated as a document, and matched according to the appropriate retrieval formula.

To match pieces of text we used the same formula for all experiments, a reliable form of the cosine measure:

$$cos(q,d) = \frac{\sum_{t \in q \wedge d}(w_{q,t} \cdot w_{d,t})}{\sqrt{\left(\sum_{t \in q}(w_{q,t})^2 \cdot \sum_{t \in d}(w_{d,t})^2\right)}}$$

with weights that have been shown to be robust and give good retrieval performance [2]: $w_{q,t} = log(N/f_t) + 1$ and $w_{d,t} = log(f_{d,t} + 1)$ where $f_{x,t}$ is the frequency of t in x, N is the number of documents in the collection, and f_t is the number of documents containing t.

To expand query, we first evaluate the original query against the database of documents or passages. The top N documents as determined by the cosine formula are then obtained. The top M terms were determined by using the formula:

$$\frac{Freq.\_in\_top\_N\_docs}{log(K + Freq.\_in\_all\_docs)}$$

Two ranked lists were combined by summing normalized scores. Scores were normalized by ensuring the top-ranked document had a score of 1.0 for each list. Thus the score becomes

$$\alpha \frac{Score1}{Max_{Score1}} + (1 - \alpha) \frac{Score2}{Max_{Score2}}$$

Unless otherwise mentioned, $\alpha = 0.5$.

3 Ad Hoc Task

Our main experiments this year have been to do a comprehensive factor analysis of most of the main contributors to successful automatic vector space retrieval. We look at stopping, stemming, passage retrieval, term expansion, methods of combination, and query length. Our methods of passage determination and methods of combination have some novel aspects but our main interest is to report on how all of the above factors interact. We performed no experiments on matching formula, or the use of adjacency information, such as phrases.

Term Experiments

Our first set of experiments were carried out on the TREC5 dataset with TREC5 queries. In these experiments we looked solely at the term definition. We built the TREC5 database 7 times where terms were defined as:

Base All valid strings (up to 256 characters long and not an SGML tag.)

Casefold Each string is converted to solely lower case letters and numbers.

Stop w/0 c.f. Base case with stop words removed.

Stop Stop words removed after casefolding.

Stem w/o c.f. Stem each string in the base case

Stem Stem after casefolding.

Stop & Stem Casefold, stop, and then stem (Standard processing.)

Having built the database, the queries were processed in the same way. The results for the description queries and full queries are given in Tables 1 and 2. The title only runs were very low but gave similar results. As can be seen, there are no surprises, and stopping and stemming English text is again shown to be appropriate.

Experiment	5docs	10docs	20docs	200docs	Average
Base	0.22	0.19	0.18	0.07	0.079
Casefold	0.23	0.21	0.18	0.08	0.087
Stop w/o c.f.	0.24	0.22	0.19	0.08	0.095
Stop	0.25	0.23	0.20	0.08	0.098
Stem w/o c.f.	0.23	0.20	0.17	0.08	0.087
Stem	0.25	0.22	0.19	0.09	0.094
Stop & Stem	0.25	0.22	0.21	0.09	0.106

Table 1: Baseline Experiments – precision for description queries 251–300

Passage and Expansion Experiments

Our next set of experiments were performed on TREC6 data using TREC6 queries. In these experiments we assume all text is stopped and stemmed and now look at the use of passages and term expansion to improve retrieval performance. These experiments were performed using the description queries. We first examine replacing the query with M terms selected from the top N documents. These expanded queries may have the original terms in them but there is no guarantee. Previous experiments suggest that one should select between 10 and 100 terms from between 10 and 100 documents [6]. In these experiments we fixed the number of terms to be 40, and used either 15 or 30 documents. K is set to 1. The next experiments looked at selecting the best document based on

Experiment	5docs	10docs	20docs	200docs	Average
Base	0.22	0.19	0.18	0.07	0.079
Casefold	0.23	0.21	0.18	0.08	0.087
Stop w/o c.f.	0.24	0.22	0.19	0.08	0.095
Stop	0.25	0.23	0.20	0.08	0.187
Stem w/o c.f.	0.42	0.34	0.27	0.12	0.163
Stem	0.42	0.38	0.32	0.12	0.178
Stop & Stem	0.40	0.39	0.32	0.12	0.192

Table 2: Baseline Experiments – precision for full queries 251–300

the best N word passage in the document. Last year a very wide range of sizes were investigated, so this year only 100, 150, and 200 word passages are reported (a wider range was investigated).

Now it is possible to expand queries using the ranking obtained from either document ranking or passage ranking. Given the improved performance of passage retrieval it may be desirable to use passages to select new terms and then rank passages using these new terms. As we see in Table 3 there is no gain obtained from this method.

Of course it is possible to merge the original query with the expanded query. We show the result of merging the original query and the expanded query taken from the top 30 passages and then matching against 150 word paragraphs. Again the result is not as good as the original passage query.

Experiment	5docs	10docs	20docs	200docs	Average
Base	0.29	0.25	0.19	0.08	0.105
Expand-15	0.31	0.25	0.21	0.07	0.106
Expand-30	0.27	0.22	0.20	0.07	0.101
Passage-100	0.32	0.28	0.25	0.10	0.174
Passage-150	0.34	0.31	0.25	0.09	0.176
Passage-200	0.33	0.30	0.25	0.09	0.167
Expand-15-150	0.28	0.26	0.22	0.08	0.134
Expand-30-150	0.29	0.27	0.24	0.09	0.140
Merge-30-150	0.29	0.29	0.26	0.09	0.159

Table 3: Passage and Expansion Experiments – precision for description queries 301–350

It was subsequently determined that the creators of the description field assumed that the title field would be used jointly. Thus the results for the corresponding runs using title and description queries

are shown in Table 4. As can be seen, there is a substantial improvement in the baseline—it is also an improvement over the equivalent baseline for title only shown in Table 8. However the big gains occur using passage retrieval. These gains are huge—about 79%. There is no gain available through expansion.

Experiment	5docs	10docs	20docs	200docs	Average
Base	0.33	0.28	0.25	0.10	0.136
Expand-15	0.30	0.28	0.23	0.08	0.110
Expand-30	0.30	0.27	0.23	0.08	0.111
Passage-100	0.45	0.40	0.33	0.12	0.242
Passage-150	0.50	0.43	0.34	0.12	0.243
Passage-200	0.46	0.42	0.34	0.12	0.238
Expand-15-150	0.36	0.33	0.26	0.09	0.157
Expand-30-150	0.32	0.30	0.28	0.10	0.161
Merge-30-150	0.32	0.30	0.30	0.11	0.194

Table 4: Passage and Expansion Experiments – precision for title+description queries 301–350

Combination Experiments

While some methods that we have described seem to give little improvement, they do provide additional evidence that may be used in combination with other factors. However, combination of evidence works well when there are different factors that give roughly equivalent performance. We consider combining 4 different pieces of evidence—the baseline run, the expansion using 30 documents, the passage run using 150 word passages, and and expansion using 30 passages of 150 words. The results are shown in Table 5.

As can be seen only small gains can be obtained above the top performing run, the passage run. By comparison, for the same set of experiments on disk2 using TREC5 queries, we found that passages gave a 20% improvement on the baseline, but that combining passages with passages on expanded queries gave another 20% gain, and that by combining all forms of evidence gave a further 10% gain. This highlights that combination of evidence works at its best when the forms of evidence are roughly comparable in quality.

The corresponding runs using title and description fields together as the query are shown in Table 6. Due to the large imbalance between the performance of the passage runs and all other runs there is no improvement over the passage runs.

Experiment	5docs	10docs	20docs	200docs	Average
Base+Exp-30	0.28	0.26	0.22	0.09	0.125
Passage-150+Exp-30	0.36	0.28	0.27	0.10	0.180
Base+Expand-150-30	0.36	0.32	0.28	0.09	0.164
Passage-150+Exp-150-30	0.34	0.32	0.28	0.10	0.170
Combine All - mds601	0.35	0.30	0.24	0.09	0.138

Table 5: Combination Experiments – precision for description queries 301–350

In Tables 7 and 8 we show the corresponding runs for full queries and title queries. The expansion algorithm was still under development at the time of these runs and is clearly problematic, particularly for passages. Unlike the description queries and title queries, there were much smaller gains made by passages for full queries.

Conclusions

Our experiments have shown yet again that stopping and stemming work well for English. We have seen that using passages gives very good gains. We have seen that expansion does not improve performance on its own. It is robust to a in terms of the numbers of document and number of terms selected, but it may be quite sensitive to the weighting formula for selecting terms. more work is needed here. When combined with an original query there is a consistent performance improvement. However if some portion of the evidence to be combined is of poor quality this can hurt performance.

Experiment	5docs	10docs	20docs	200docs	Average
Base+Exp-30	0.36	0.30	0.26	0.11	0.147
Passage-150+Exp-30	0.44	0.37	0.32	0.12	0.214
Base+Expand-150-30	0.40	0.37	0.32	0.12	0.188
Passage-150+Exp-150-30	0.40	0.38	0.33	0.12	0.229
Combine All	0.40	0.37	0.32	0.12	0.204

Table 6: Combination Experiments – precision for title+description queries 301–350

Experiment	5docs	10docs	20docs	200docs	Average
Base	0.45	0.38	0.32	0.13	0.196
Exp-30	0.32	0.29	0.28	0.10	0.142
Passage-150	0.45	0.38	0.32	0.13	0.200
Expand-150-30	0.27	0.25	0.21	0.08	0.090
Combine All - mds602	0.42	0.37	0.32	0.13	0.230

Table 7: Combination Experiments – precision for full queries 301–350

4 Chinese Retrieval

First Experiments

For our baseline experiments, we tried indexing each document on characters, words, and bigrams. For character indexing we treated each document as a series of distinct characters, and used these to build our index. Queries consisted of all the characters in the document. For word indexing we parsed documents into words using an online dictionary kindly made available by the Berkeley group. We used *greedy parsing*, in which we matched the longest entry in the dictionary at any point. Although this is not the best strategy, it works reasonably well. For bigrams we used every possible pair of adjacent characters that did not include punctuation. For example a sequence of 7 Chinese characters *abc.def* would generate pairs *ab, bc, de, ef* but not *c., .d, or cd.* We used the *mg* system for our experiments. In calculating query/document similarity we used the standard cosine measure. Results for these experiments are shown in Table 9.

Mutual Information

In this experiment we were interested in seeing how well we could segment the text into words without the use of a dictionary, but rather relying on the mutual information contained in the corpus. This

Experiment	5docs	10docs	20docs	200docs	Average
Base	0.28	0.25	0.20	0.09	0.127
Exp-30	0.24	0.22	0.19	0.07	0.087
Passage-150	0.40	0.33	0.28	0.10	0.168
Expand-150-30	0.22	0.21	0.18	0.05	0.076
Combine All - mds603	0.31	0.29	0.26	0.10	0.157

Table 8: Combination Experiments – precision for title queries 301–350

Experiment	5docs	10docs	20docs	200docs	Average
Characters	0.73	0.72	0.69	0.33	0.455
Words	0.78	0.76	0.74	0.35	0.498
Bigrams	0.74	0.72	0.68	0.35	0.494

Table 9: Baseline Experiments – precision for queries 29–54

is similar to the approach used by [3], based on the mutual information idea proposed by [5]. Mutual Information is defined as $I(x,y) = log_2 \frac{p(x,y)}{p(x)p(y)}$ where $p(x)$ and $p(y)$ is the probability of occurrence of characters x and y respectively in the corpus, while $p(x,y)$ is the probability of the two characters occurring together. If the two characters are related the value of $I(x,y)$ will be high, suggesting that the bigram xy may in fact be a word. Thus a two step method is needed. First frequencies are determined from the source text, and bigrams with a mutual information value above a threshold are presumed to be words. We used $I(xy) = 7$ as the threshold, the same as [5]. Second the text is parsed, sentence at a time using the words so gathered. Our results gave an average precision of 0.302 on queries 1–28, which is inferior to the 0.374 figure reported by [3].

Experiment	5docs	10docs	20docs	200docs	Average
Mutual Information	0.76	0.75	0.74	0.35	0.462

Table 10: Mutual Information – precision for queries 1–28

Expansion of characters and words

In these experiments we implemented a simple form of feedback by using the top 30 documents returned for each query in experiments one and two. We did a frequency count of the terms in these documents and ranked them using $\frac{f_i}{log_2(df_i+20)}$ where f_i is the frequency of the ith term the top 30 documents, and df_i is the document frequency. This effectively selects relatively infrequent words, but avoids the undue influence from very rare words by adding 20 to the denominator. Results for these expanded queries are shown in Table 11. Only in the case of bi-grams did the expansion give any improvement over the equivalent baseline. We think that further refinement of our expansion mechanism is required, and are currently investigating this.

Experiment	5docs	10docs	20docs	200docs	Average
Characters	0.73	0.72	0.69	0.33	0.455
Exp-Characters	0.68	0.65	0.54	0.20	0.239
Words	0.78	0.76	0.74	0.35	0.498
Exp-Words	0.82	0.80	0.73	0.31	0.437
Bigrams	0.74	0.72	0.68	0.35	0.494
Exp-Bigrams	0.83	0.77	0.73	0.34	0.506

Table 11: Expanded Queries – precision for queries 29–54

Combination of Evidence

Following the hypothesis that combination of evidence from a number of sources usually improves results, we decided to try combining some of the results of our previous experiments as a final step. We tried values of 0.33, 0.5, and 0.67 for α to test the sensitivity of combination. As well as combining results from two experiments, we found that by again combining the results of these runs we could further improve performance. Our best run on the queries 1-26 was the result of first combining bigrams and expanded bigrams, then combining words and expanded words, both with $\alpha = 0.5$, then combining the results of these two runs using $\alpha = 0.33$. Results of some other combinations are shown in Table 12. Clearly combination of evidence improves retrieval effectiveness.

Exp.	Method A	α	Method B	5docs	10 docs	20docs	200docs	Average
	Bigrams	0.33	Exp-Bigrams	0.87	0.81	0.76	0.37	0.546
mds607	Bigrams	0.5	Exp-Bigrams	0.88	0.82	0.76	0.37	0.547
	Bigrams	0.67	Exp-Bigrams	0.84	0.80	0.74	0.37	0.539
	Words	0.33	Exp-Words	0.88	0.84	0.79	0.36	0.528
	Words	0.5	Exp-Words	0.86	0.83	0.80	0.36	0.535
	Words	0.67	Exp-Words	0.86	0.83	0.77	0.36	0.536
	Bi/BiX-0.5	0.5	Word/WordX-0.5	0.88	0.84	0.78	0.37	0.560
mds608	Bi/BiX-0.33	0.67	Word/WordX-0.67	0.87	0.82	0.77	0.38	0.560
mds609	Bigrams	0.50	Word Exp	0.85	0.83	0.77	0.37	0.548

Table 12: Statistics of Chinese text collections – precision for queries 29–54

Conclusions

We have seen that using bigrams as a basis of retrieval is quite effective and provides a simple low-cost solution to effective Chinese retrieval. Term expansion and combination of evidence can be used to improve retrieval effectiveness, however we need to do more work to understand how to apply them well in the context of Chinese IR.

5 Interactive Retrieval

Goals

The high-level goal of the Interactive Track in TREC-6 is the investigation of searching as an interactive task by examining the process as well as the outcome. In particular, the task set for the interactive track was the investigation of multi-aspect queries. The RMIT interest in the TREC-6 interactive track was twofold. The primary interest was to develop an interactive system that would use feedback from user to user to cluster and re-rank relevant documents. The secondary interest was to develop an interface to aid users to structure and organize candidate documents in order to compose answers to information needs.

Our goals in this project were therefore three-fold. Firstly, to develop an interactive system that could be paired experimentally with the ZPRISE control system. Secondly, to use that system to interactively cluster and re-order candidate documents to better allow the user to identify documents of interest. Thirdly, to extend that system to permit the user to organize candidate documents or passages such that a structured answer to their information need might be formed.

RMIT WWW/MG system

This was our first attempt to be involved in the TREC interactive track, and, unfortunately, we were only able to complete a partial WWW-based prototype for the experimental phase. Per the interactive track model, we ran four subjects on the ZPRISE control system, and on an WWW/MG-based system. These results can best be considered a comparison of ZPRISE with interactive MG, rather than an evaluation of a new experimental system. Work is continuing on the implementation of the full interactive prototype.

The WWW/MG prototype system was intended to mimic the functionality of the control system. Similarly to the control, it allowed users to issue free text queries, resulting in a list of candidate documents (matching documents identified by MG), but with the addition of the ability to identify

and label each aspect found within candidate documents from within the document presentation interface. This can be seen in Figure 1. The prototype was implemented by building a MG database from the FT test collection. The HTML interface was dynamically generated by a set of CGI Perl and JavaScript programs, allowing users' queries to be converted into an MG query, the results of which were then parsed and a synopsis converted to HTML. Requests to view specific documents were also passed to MG, and the resultant document text displayed as HTML. Additional tasks such as tracking user judgments, user-determined topic aspects, and ancillary logging were also performed.

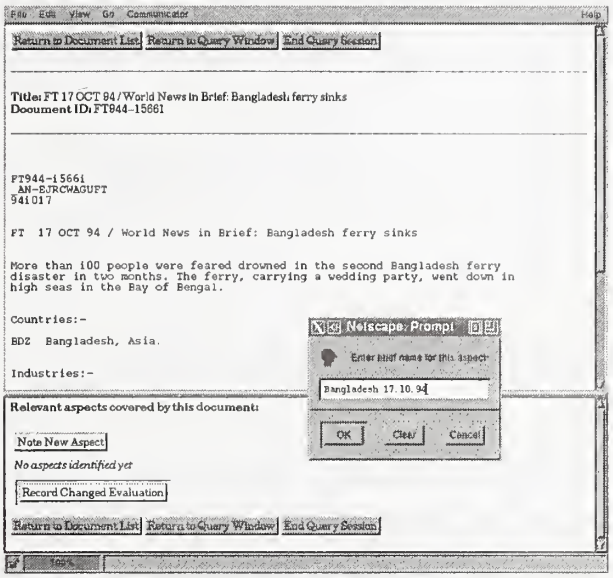


Figure 1: Document Viewing Interface

As the prototype system comprised a relatively simple interface to an existing retrieval system and was intentionally similar to the control system in design, it is not surprising that the experimental results were close to the control results. For the control systems, subjects identified aspects at an average precision of 0.779 and an average aspectual recall of 0.499, in, on average, approximately 17 of the 20 minutes available. For the experimental system, subjects identified aspects at an average precision of 0.805 and an average aspectual recall of 0.466, in, on average, approximately 17 of the 20 minutes available.

As part of our analysis of the results we considered two questions: what agreement was there on aspectual relevance, and what agreement was there on what the aspects were. There was a great deal of disagreement between experimental subjects and the NIST assessors (Table 13). Of the documents considered relevant by the pool of experimental subjects, over half were rejected by the NIST assessors as irrelevant. Conversely, numerous documents considered relevant by either or both the NIST assessors and subjects from other sites were viewed by RMIT experimental subjects and

rejected as irrelevant. This occurred both when using the WWW/MG system, and when using the ZPRISE control system. From a local perspective, the same phenomenon occurred: on several occasions, for the same queries, the same documents were viewed by separate subjects, only for one to decide the document contained relevant aspects, and the other that it did not.

Query	Subject Relevant	NIST aspects	NIST Relevant	NIST Irrelevant
303i	22	7	5	17
307i	103	23	54	49
322i	63	9	21	42
326i	47	9	31	16
339i	28	10	7	21
347i	86	26	43	43
Totals:	349	84	161	188

Table 13: Relevant and irrelevant documents

Tables 14 and 15 provide a comparison of the aspects found (or not found) by WWW/MG searchers for queries 303i and 339i.

Table 14 reveals a marked difference between the aspects nominated by the NIST assessors and those listed by the WWW/MG experimental subjects. This is evidenced by the aspects to be found in document FT924-286: the NIST assessor discovered four distinct aspects of the topic, whereas the RMIT subject indicated it covered just a single aspect, "black hole study". Other interesting differences include document FT934-54181, which the NIST assessor described as representing the aspect "generally good, better, better than expected results", but which was viewed and discarded as not relevant by the RMIT subject; and document FT941-17652 for which the reverse was true.

Table 15 has two interesting characteristics. One is that experimental subjects are not accurate readers! Searcher s2 noted that the document was relevant to two aspects, but missed three others; whilst searcher s4 noted three aspects, but not two others. (Searcher s2 had not previously seen documents containing relevant aspects; searcher s4 had previously seen and saved a document containing aspect 4, but not aspect 6.) The other is the difficulty of defining relevance: subjects indicated that the documents FT933-5910 and FT942-17255 contained aspects relevant to the topic, but were rejected as irrelevant by the assessor.

We have seen that in Query 303i there is a very significant difference in how the "intellectual space" is divided into aspects. The consequence is that there can be a very large difference in precision and aspectual recall as a result. In Query 339i there is less issue of the nature of the aspects, but even if subjects agree on relevance, they did not recognise the presence of some aspects, affecting their

"personal aspectual recall", but not the system evaluated performance.

NIST aspects	
1	has inspired new cosmological theories
2	study of gravitational lenses
3	more precise estimate of scale, size, and age of universe
4	picture of more distant galaxies/objects
5	generally good, better, better than expected results
6	contradicted existing cosmological theories
7	supported existing cosmological theories

RMIT aspects	
a	black hole study
b	images
c	Hubble,wonder image, universe theory
d	origin of universe hubble, fall of contemporary cosmology theories

NIST aspects	Document	RMIT aspect
1, 2, 3, 4	FT924-286	a
5	FT944-15661	b
5	FT934-54181	—
—	FT941-17652	c
3, 4, 5, 6, 7	FT941-17652	d

Table 14: Query 303i: NIST and RMIT aspects

Concluding remarks

We are pleased to have taken part in this year's interactive track. It has raised some philosophical and methodological issues of interest and concern. We plan now to continue development of an interactive prototype, utilizing clustering and feedback to group and re-order the pool of candidate documents dynamically, leading to a system designed to support analysis and synthesis of structured answers to information needs.

NIST aspects		RMIT aspects	
1	Alcav	1	alcar
2	pivacetam	2	piracetam
			Piracetam of UCB Belgium
3	oxiracetam	3	oxiracetam
			oxiracetam of SmithKline Beecham
4	tacrine - Cognex	4	cognex
			Warner-Lambert, Cognex,
			Good evaluator of effect of drug to alz
5	physostigmine	5	physostigmine of Forest Lab US
6	Aviva	6	aviva
7	velnacrine - Mentane	7	velnacrine (Mentane) of Hoechst Germany
8	selegiline (Eldepryl)	8	selegiline of Sandoz Switz
9	Zofran (ondansetron)	9	(not found)
10	denbuflline	10	denbufylline
		a	silicon intake to prevent alz
		b	tacrine

NIST aspects	Document	RMIT aspect
1, 3, 10	FT922-1565	1, 3, 10
2, 3, 4, 5, 6	FT922-715	searcher s2: 2, 6
		searcher s4: 2, 3, 5
4	FT924-8306	4
4	FT931-2434	4
4, 7, 8	FT932-7262	7, 8
—	FT933-5910	a
—	FT942-17255	b

Table 15: Query 339i: NIST and RMIT aspects

6 Speech Retrieval

We participated in the full SDR track. Our speech experiments explored the use of phoneme sequences as matching units instead of words. Phonemes were extracted from the speech tracks and triphones created to perform retrieval. For comparison, the transcripts were also translated to phoneme sequences. The translation to phonemes used the Ainsworth algorithm [1].

MG, developed at RMIT and the University of Melbourne, is the retrieval engine used. The recog-

nition engine used is HTK which is developed at Cambridge.

The reference experiment consisted of retrieval based on the textual documents provided (LTT). The first baseline experiment used the transcribed documents provided by IBM (SRT). For both experiments, documents and queries were stemmed and casefolded. In addition, the queries were stopped. The average length of the queries was 5 words. Results for the reference run is shown as MDS612 and the first baseline run is shown as MDS613 in Table 16.

The second baseline experiment investigated the performance of phoneme retrieval when recognition is assumed to be perfect. The documents from the reference run were translated to phonemes and then transformed to triphones before indexing. The queries were not stopped prior to the transformation. Results for this run are shown as MDS615.

We used triphones because it has a higher noise tolerance than words. In addition, word boundaries becomes less important.

A simple phoneme model was built for 61 phones. The phoneme model was trained using the speech training documents provided. The training data did not contain explicit details about phone boundaries within and between words. Explicit segment and section times were provided for the training process instead. We had about 1400 test and 500 training documents. For our experiments, most of the longer training and test documents were not used. Initial recognition results indicate about 16% recognition accuracy of the phonemes. A reason for such poor performance may be due to the lack of information on phoneme boundaries. There may also be an error in the training which we have yet to find.

Post-trec experiments included the addition of another 86 documents from the i-disk (i960606, i960610, i960611). These were excluded because of difficulties in processing the larger speech documents. The reference experiment was repeated as well as the second baseline run. The results are shown as ref-new-s and ref-new-ph-l respectively. The stoplist may have been too aggressive for this document collection. The results of queries which were not stopped are shown as ref-new-l. Some of the stop terms were useful in retrieving relevant documents. This was indicated by an improvement in mean reciprocal.

The transcribed documents (SRT) were translated and transformed to triphones. Results are shown as srt-new-ph-l. The effects of an imperfect recogniser contributed to the retrieval of many irrelevant documents.

For triphone retrieval, important textual terms were lost but retrieval was possible using part of the terms at triphone level. It was found that unimportant terms at the word level became important at the phoneme level. For example, the term classic is translated to klasik which is transformed to "... kls las asi sik ...". The triphone "las" helps retreive the relevant document.

Experiments have shown that without being given more explicit boundary information, the recognition result can be improved to approximately 30%. This was accomplished by segmenting the training documents into smaller segments of about 30 seconds. However, this recognition model resulted in degraded retrieval effectiveness. This could mean the model is not recognising at all or the test documents have to be segmented as well.

	Mean Rank	Mean Reciprocal
MDS612	5.31	0.7036
ref-new-s	5.48	0.6899
ref-new-l	13.10	0.7238
MDS613	10.11	0.5207
MDS614	229.20	0.0046
MDS615	8.71	0.7316
ref-new-ph-l	11.47	0.7340
srt-new-ph-l	23.49	0.5472

Table 16: Full Speech Experiments

References

[1] William A. Ainsworth. A system for converting English text into speech. *IEEE Transactions on Audio and Electroacoustics*, AU-21(3):288–290, Jun 1973.

[2] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, pages 292–300, Dublin, Ireland, July 3–6 1994. Springer–Verlag.

[3] A. Chen, J. He, L. Xu, F.C. Gey, and J. Meggs. Chinese text retrieval without using a dictionary. In N. Belkin, D. Narasimilau, and P. Willett, editors, *Proceedings of the 20th Annual International Conference on Research and Development in Information Retrieval*, pages 42–49, Philadelphia, U.S.A., August 27–31 1997. ACM.

[4] J.B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computation*, 11(1–2):22–31, 1968.

[5] R. Sproat and C.L. Shih. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351, 1990.

[6] R. Wilkinson. Using combination of evidence for term expansion. In J. Furner and D.J. Harper, editors, *Proceedings of the BCS-IRSG 19th Annual Colloquium on IR Research*, Aberdeen, Sotland, 1997. Springer-Verlag.

Appendix A – Stopwords

0 1 2 3 4 5 6 7 8 9 a about above across after afterwards again against all almost along already also although always am among amongst an and another any anybody anyhow anyone anything anywhere ap apart are around as at b be became because become becomes been before beforehand behind being below beside besides best better between beyond both but by c can cannot cant co could d described did do does doing done down during e each eg eight eighth either else elsewhere end ended ending ends enough et etc even evenly ever every everybody everyone everything everywhere ex except f far few fifth first five for four fourth from furthermore g great greater greatest h had has have having he hence her here hereafter hereby herein hereupon hers herself high higher highest him himself his hither how howbeit however i ie if in inasmuch indeed insofar instead into inward is it its itself j just k l large largely last later latest latter latterly least less lest long longer longest m many me meanwhile more moreover most mostly mr mrs much my myself n namely neither never nevertheless newer newest next nine ninth no nobody non none noone nor not nothing now nowhere o of off often oh old older oldest on once one ones only onto or other others otherwise our ours ourselves out over overall p per perhaps possible q que quite r rather really s same second secondly self selves seven seventh several shall she should since six sixth small smaller smallest so some somebody somehow someone something sometime sometimes somewhat somewhere still such t than that the their theirs them themselves then thence there thereafter thereby therefore therein thereupon these they thing things third this those though three through throughout thru thus ten tenth to together too toward towards turn turned turning turns twice two u under unless until unto up upon us v very via viz vs w was we were what whatever when whence whenever where whereafter whereas whereby wherein whereupon wherever whether which while whither who whoever whole whom whose why will with within without would x y yet you your yours yourself yourselves z zero

Verity at TREC-6: Out-of-the-Box and Beyond

Jan O. Pedersen, Craig Silverstein, Christopher C. Vogt
Verity, Inc.
894 Ross Dr.
Sunnyvale, CA 94089
jpederse@verity.com, csilvers@cs.stanford.edu, vogt@cs.ucsd.edu

February 17, 1998

Abstract

The Verity Trec-6 entry focused on the performance of the built-in search facilities of the commercially available Verity engine and explored the impact of simple enhancements. The *ad hoc* results show that considerable improvements can be achieved through the application of standard and more experimental techniques. The *routing* results show that respectable performance can be achieved simply through careful parameter tuning.

1 Introduction

The focus of Verity's TREC-6 work was to measure the performance of the built-in Verity retrospective search facilities and to explore simple extensions which have the potential to improve performance. Participating in both the *ad hoc* (short query, automatic only) and *routing* Category A tracks, we investigated both standard techniques (e.g., IDF, length normalization, pseudo-feedback, and training routing classifiers on a per-query basis) along with more experimental ones (linguistic parsing, clustering, local region training, and source-sensitive training). None of these are yet built into the Verity engine, although they can be implemented on top of the Verity Query Language. As our routing experiments show, the built-in Verity QBE facility can perform on a par with other TREC entries. Our ad hoc experiments show, as expected, that adding the "standard" techniques boosts performance significantly, whereas most of the other techniques also boost performance, but to a lesser degree.

Our *ad hoc* approach began with a very simple baseline system and progressively added techniques, all built on top of the Verity Query Language (VQL). Each new technique except clustering successfully improved performance. Our *routing* approach emphasized using queries automatically generated by the built-in VQL Query-By-Example facility, tuning the parameters to that facility based on training examples. Thus, our *routing* approach represents what a current Verity user could achieve without adding to built-in functionality, but our *ad hoc* approach represents what VQL is capable of achieving.

All experiments were performed on a Sparc Ultra Enterprise 3000 with four 250 MHz UltraSparc processors and 2 GB of main memory, running Solaris 2.5.1. The experiments were run using release 2.2 of Verity's search engine.

2 The Ad Hoc Task

The Verity search engine supports a rich, fuzzy Boolean query language. However, it is intended for use with hand-crafted queries, not TREC-style natural language queries. This is not a functional limitation; one can supply a query analyzer that converts natural language queries to fuzzy Boolean. However, the built-in Verity query analyzer is missing some components typically used in tandem with natural language queries. In particular, it employs a non-linear scoring scheme optimized for short queries that saturates (achieves a maximum score) too rapidly to be useful for Trec-length queries.

Nevertheless, the underlying Verity query language supports features that allow a state-of-the-art natural language query analyzer to be built on top of the Verity engine. Query words may be weighted arbitrarily, allowing IDF (inverse document frequency) information to be encoded in the query. The weighting scheme may be changed to a linear vector model, ameliorating saturation problems. Summarization facilities provide a novel method for length normalization and also a basis for performing pseudo-feedback. Finally, we can use the ability to cluster result sets to provide a refinement of the typical pseudo-feedback algorithm.

For the *ad hoc* TREC-6 task, we started with the baseline query analyzer that converted each natural language query into a sum of query terms. Then we added each of the following units, in turn, on top of the base analyzer:

1. Phrase identification and part-of-speech-based word elimination;
2. Word elimination based on shallow syntactic analysis;
3. IDF weighting;
4. Length normalization;
5. Pseudo-feedback; and
6. Clustering.

Notice that each of these steps involves merely modifying the query, or, in the case of length normalization, modifying the corpus. Thus they can all be implemented without having access to the internals of the underlying search engine.

The results of these modifications are summarized in Table 1, which looks at the precision at 30 documents for both TREC-6 and TREC-5. Since the algorithms build on each other, we show both the improvement in performance over the previous algorithm as well as over the base algorithm described in Section 2.3. In the following sections, we study each refinement in turn.

2.1 Experimental Setup

The TREC-6 *ad hoc* test bed consists of about 2.5 gigabytes of data, in about 550,000 documents, from the Congressional Record, Financial Register, Financial Times, Foreign Broadcast Information Service, and LA Times collections. This is augmented with 50 topics along with relevance judgements. Each topic consists of three fields: a title (2-3 words), a description (1-2 sentences), and a narrative (a paragraph listing specific criteria for accepting or rejecting a document). We examined only the description of each topic, ignoring the other fields.

For each algorithm studied, we ran the algorithm on each query. The algorithm attached a score to each document based on the query and retrieved the top 1000 documents, sorted by score. We studied two major statistics based on the retrieved document set for each query: the total recall

trec6	prec30	Δprev	Δbase
base	0.0700	—	—
+phrase	0.0987	40%	40%
+syntax	0.1087	10%	55%
+idf	0.1727	58%	146%
+len	0.2207	27%	215%
+feedback	0.2473	12%	253%
+cluster	0.2333	−5%	233%
out-of-box	0.0860	—	22%
base+idf	0.1247	—	78%

trec5	prec30	Δprev	Δbase
base	0.0673	—	—
+phrase	0.1327	97%	97%
+syntax	0.1447	9%	115%
+idf	0.2027	40%	201%
+len	0.2113	4%	213%
+feedback	0.2493	17%	270%
+cluster	0.2513	0%	273%
out-of-box	0.0800	—	18%
base+idf	0.0913	—	35%

Table 1: The performance of various refinements to the baseline Verity search engine. Performance is given in terms of the precision at 30 documents, averaged over all 50 queries in TREC-6 (left table) and TREC-5 (right table). Relative performance is in terms of both the previous algorithm, which the current algorithm builds on, and the original, baseline algorithm. Algorithms below the line are not in the main chain of refinements and are used to examine the effectiveness of certain refinements in isolation, for instance IDF without term selection.

(that is, how many of the 1000 retrieved documents were judged relevant), and the precision at 30 documents (that is, how many of the top 30 retrieved documents were judged relevant).

2.2 The Out-of-the-Box Algorithm

The Verity `<FREETEXT>` operator is intended to handle short natural language queries. This built-in, out-of-the-box algorithm performs poorly over the relatively long TREC-6 queries, as Table 1 attests. In the table, the out-of-the-box algorithm yields only a small improvement over the baseline algorithm. It also performs significantly worse than the "phrase" algorithm, which performs similar term selection. This disparity can be attributed to the scoring function, which, as we mentioned in Section 2, saturates with many query terms. The out-of-the-box algorithm uses this default scheme, while the baseline algorithm uses an alternate scheme that does not have the saturation problem. We conclude that even with relatively aggressive term selection, many documents contain a large number of query terms for most queries, causing score saturation to be an issue.

2.3 The Baseline Algorithm

While the freetext parser is helpful for natural language queries, it is not truly a baseline, at least in the context of the Verity search engine, because it performs syntactic parsing of the query before invoking the search routines. We therefore, instead, build our algorithms on top of the following, simple, baseline algorithm:

1. Considering every word (including stop words) as a separate term in the query, weight each word equally so the weights sum to 1.

2. Combine the query terms using the `<SUM>` operator. Pass this query to the Verity search engine.

The `<SUM>` operator determines a document's score by summing the scores of each query term on that document. The score of a term on a document is determined as follows: A term has score 0 if it does not occur in the document, and a score equal to its weight if it occurs (logically) an infinite number of times in the document. The score is in between if it occurs a finite number of

times in the document, but it quickly grows to the weight. The Verity engine performs stemming on each query term. It is case sensitive only if the query term includes an upper-case letter. (In our experiments, we keep the case of the original TREC-6 description, except we lower-case the first word of each sentence.)

This equal weighting scheme is clearly a bad idea without careful term selection, and indeed the baseline algorithm performs quite badly. Its performance can be seen as the bottom curve in Figure 1 and the first row in Table 1.

2.4 Term Selection I: Phrase and Part-of-speech

One obvious way to improve the baseline algorithm is to perform term selection. In this step we perform a similar analysis as is done by the freetext parser. However, instead of the built-in parser we use ENGCG, an off-the-shelf parser sold by the Lingsoft Corporation (http://www.lingsoft.fi) We use ENGCG because the built-in parser cannot perform the syntactic analysis we do in the next section, and to use the ENGCG parser there we must use it here as well. This parser performs the following functions:

1. Delete words due to their part of speech. Nouns, verbs, adjectives, adverbs, interjections, numerals, abbreviations, and participles are retained. Coordinating conjunctions, subordinating conjunctions, determiners, infinitive markers, prepositions, and pronouns are removed.

2. Collect words into phrases, when appropriate. There are two types of phrases: noun phrases (when one or more nouns modify a noun) and adjective phrases (when one or more adjectives modify a noun or noun phrase).

3. Weight each term or phrase so the weights sum to one. Terms are weighted equally, but adjective phrases are given twice the weight, and noun phrases three times the weight, of terms.

4. Combine the query terms using the <SUM> operator. Pass this query to the Verity search engine.

Identification of phrases is possible because ENGCG is a shallow syntactic parser, so it can determine which word a modifier modifies. If A modifies B, everything between A and B is taken to be a phrase, and the type of phrase is determined by the part of speech of A. Thus, "Food and Drug Administration" is a noun phrase.

Phrases are weighted highly purely due to experimental evidence that it improves recall. This is expected, since previous research has shown that phrases are better markers of relevance than individual terms.

Recall that individual terms are scored based on their frequency in a document. Phrases are scored somewhat differently. A phrase gets full credit if the phrase appears as consecutive words in the document. However, it can get up to half credit if the words in the phrase appear in the document, but not consecutively. Each phrase component found contributes equally to the score. Furthermore, if the words in the phrase appear close to each other but not consecutively, up to 75% credit is possible. This accommodates situations where the phrase is interrupted by a modifier or interjection. Experimental results show that this complicated scoring system improves on the naive scoring system based on considering the phrase merely as a single word with internal spaces.

Term and phrase selection improve the performance dramatically, as can be seen in Figure 1 and the second row in Table 1.

2.5 Term selection II: Shallow Syntactic Parsing

The ENGCG parser is a shallow linguistic parser, which means it can identify phrases, clauses, and the role words play in a sentence (for instance, whether a noun is the subject, whether a verb is transitive or intransitive, and so on). We use this to recognize patterns that are often found in natural language queries, and to delete uninformative terms in the query.

For example, we can detect if the verb is imperative ("Find all documents relating to slavery in Brazil"), in which case the object of the verb is merely a placeholder like "documents." In this case we delete the verb and its object. After part of speech elimination, we will be left with the query `<SUM>(slavery, Brazil)`.

As another example, if there are chained prepositional phrases starting with "of," only the last prepositional phrase is kept. Thus, in sentences like "The exportation of some part of U.S. industry," we would throw out "part." In every query we have examined, the first "of" prepositional phrase holds a contentless word.

This yields only a small addition to the algorithm:

1. Delete words according to their syntactic function.
2. Analyze the remaining words using the algorithm in Section 2.4.

As Figure 1 and the third row in Table 1 demonstrate, this more aggressive term selection improves the quality of the result set, though not dramatically. Since these rules are only applied to specific query forms, the effect of this step is pronounced in some queries but non-existent in most. See Tables 2 and 3 for a per-query analysis.

2.6 Term Weighting

There were two weaknesses of the baseline algorithm: lack of term selection and lack of term weighting. Previous algorithms have attacked the first of these problems; now we approach the second.

One very popular term weighting scheme is tf.idf weighting, that is, weighting by term and inverse document frequency. The Verity engine performs some variant of term weighting by default, so we need only to add IDF weighting.

The IDF weight of word w is defined to be

$$\log \frac{n}{n_w}$$

where n is the total number of documents and n_w is the number of documents containing word w. While the Verity search engine does not support IDF explicitly, it does allow us to query how many documents a term occurs in. Therefore, we can, in a preprocessing step, determine the IDF weight of every word (and phrase) in the query. This yields the following algorithm:

1. Perform term selection as in Section 2.5.
2. Weight each term/phrase so that the weights sum to 1. However, instead of equal weight, each element gets a weight related to to its IDF weight. In particular, terms are weighted proportional to their IDF score, adjective phrases proportional to twice their IDF score, and noun phrases proportional to three times their IDF score.
3. Combine the query terms using the `<SUM>` operator. Pass this query to the Verity search engine.

Since phrases tend to have high IDF weights due to their relative rarity, it may seem no longer necessary to give phrases a bonus of double or triple weight. However, experiments showed that recall was improved when these bonuses were applied.

IDF weighting is also applied inside phrases. Remember that a document could get a partial score for a phrase if its constituent words occurred, even if the phrase did not. The partial score for a given constituent word is made proportional to the IDF score of that word. Thus, in the phrase "Food and Drug Administration," a document would get little bounce from having "and" but a larger jump for having "Administration."

As is consistent with previous research, IDF improved the quality of the result set tremendously (see Figure 1 and the fourth row in Table 1).

IDF is often seen as a replacement of, or improvement on, term selection. Thus, we would expect IDF to give a greater bounce to the baseline algorithm than to the term selection algorithm. As we see in Table 1, however, this is not the case. The last row of the table shows the improvement over baseline due merely to IDF: 78% for TREC-6 and 35% for TREC-5. The "+idf" row shows the advantage of adding IDF to phrase selection: 58% and 40%, respectively. For each data set, the improvement is similar in both cases. This seems to indicate that IDF and term selection actually complement each other. IDF is useful for queries with many nouns and other words term selection algorithms tend to see as identical. On the other hand, IDF may highly weight words that are tangential to the query (such as "documents" in "Find documents that relate to the crime in Europe") that term selection techniques can identify. Thus, despite some overlap in functionality overlap, IDF and term selection are complementary techniques.

2.7 Length Normalization

Collections with both short and long documents, such as the data set used for TREC-6, provide a particular problem for search engines. A canonical problem document is the dictionary, which matches every query word but is relevant to few queries. Traditionally this has been handled via length normalization, which penalizes long documents. Approaches have ranged from automatically splitting long documents to downweighting documents based on their length.

For these experiments we tried a different approach. The Verity search engine includes a summarizer, which ranks each sentence of a document based on its perceived relevance to the document's content. We chose a length (approximately 16 Kbytes, the median length of a document in TREC-6), and used the summarizer to bring documents larger than the cutoff length down to that length in size. We did this by selecting sentences in order of relevance until the cutoff was reached. Here is the resulting algorithm:

1. Replace each document in the corpus by its summarized version. Documents under the cutoff length (16K) remain unchanged.

2. Perform the algorithm of Section 2.6 on the summarized corpus.

This technique behaves differently than other length normalization techniques. If the document logically consists of many parts, one of which is much smaller than the others, then the summarizer is likely to ignore the small part entirely. Thus the search algorithm will fail on queries that match the small section. On the other hand, if the document contains many "throw-away sentences" tangential to the main thrust of the document, the summarizer will do well by ignoring the irrelevant sentences.[1] In our experiments, the summarizer did as well as, and even slightly better than, length

[1]While this might generally be considered a feature, it is not rewarded behavior in the TREC framework, since TREC judges a document to be relevant if *any* portion in the document fits the relevance criteria (as spelled out in the narrative).

normalization based on Singhal's method.

The results for the algorithm with length normalization added can be found in Figure 1 and the fifth row in Table 1.

2.8 Pseudo-Feedback

Pseudo-feedback is a common "accelerator" technique, piggybacked on other techniques. If the original technique has "good" precision, pseudo-feedback improves it; if the precision is "bad," pseudo-feedback usually makes things worse. The reason for this behavior (and one reason "good" and "bad" are not well defined concepts) is that pseudo-feedback augments the query based on terms found in highly ranked documents. If these documents are actually relevant to the query, the new query may better specify the information need, but if they are random documents the new query will be overwhelmed with irrelevant terms.

We use the Verity summarizer feature for pseudo-feedback. Just as for length normalization, the summarizer identifies sentences it deems represent the document, here we have the summarizer identify words it finds relevant. We had the summarizer identify the 5 most representative terms and phrases from each of the 20 documents deemed most relevant (according to the algorithm of the previous section).[2] We then ranked the 100 terms/phrases based on their IDF score, strongly upweighting terms recommended by more than one document. We took the 5 highest scoring terms and used them to augment the query, and performed the algorithm from Section 2.7 on the new query. Formally:

1. Perform the algorithm of Section 2.7 on the query.
2. For the top 20 documents (or for all documents with score at least 0.3, whichever is fewer), use the summarizer to find the 5 most relevant words. Rank the 100 words based on tf.idf weighting. Collect the 5 highest ranking words.
3. Form a new query by adding the 5 words from the previous step to the end of the existing algorithm.
4. Perform the algorithm of Section 2.7 on the new query.

This is a two-step algorithm. Previous research used many more than 20 documents, and many more than 5 terms, for pseudo-feedback. However, these were the parameters that gave the best results in our tests. One reason may be that for many TREC queries, the number of relevant documents is small, so it is advantageous to use few documents for pseudo-feedback. Also, our method of augmenting the query favors using few feedback terms. If we use many terms, with short queries — and many queries are short after term selection — the feedback terms will outweigh the original query terms. This is especially true since the feedback terms are chosen based on their high IDF weight.

Figure 1, and the sixth row in Table 1, show the effect of pseudo-feedback on retrieval quality. A look at the individual queries shows that it helps significantly on some queries and hurts on others; the performance on any given query is highly dependent on the quality of the five terms chosen for feedback.

[2]We actually used fewer than 20 documents if most documents had low scores. Recall that each document gets a score from 0 to 1. We only used documents for feedback whose score was at least 0.3.

2.9 Cluster-based Pseudo-Feedback

One way to judge whether the original retrieval algorithm is "good" or "bad" on a query — that is, whether performing pseudo-feedback would improve or degrade the quality of the result set — is whether the top documents retrieved have similar content. If they do, this content is likely to be related to the query. If not, the search engine likely retrieved random documents. This reasoning is called the *Cluster Hypothesis*.

Verity provides a clustering routine that we can use to judge whether a set of documents has similar content. We cluster all 1000 retrieved documents into 5 clusters. We then count how many of the top 20 documents fall into the same cluster. If it is at least half, we judge the result set coherent and perform pseudo-feedback. If not we return the result set without performing feedback.

We actually use a slightly different technique that guarantees we always perform feedback. For every i between 1 and 20, we mark i if at least half of the i top documents are in the same cluster. We then take the largest i that is marked. Since $i = 1$ is always marked, we always take at least one document for pseudo-feedback.[3]

The results of this cluster-based algorithm are shown in Figure 1 and the last column above the line in Table 1.

2.10 TREC-5 and TREC-6 results

Here we show the results for the various algorithms, both on the TREC-5 collection and the TREC-6 collection. TREC-5 can be thought of as the "training" set, since the parameters of the various algorithms (such as the weighting bonus of phrases, the 0.3 score cutoff for pseudo-feedback, and the 16K length cutoff for length normalization) were optimized for TREC-5. We see, comparing the two tables in Table 1, that in TREC-6 most methods provide a smaller increase, percentage-wise, than in TREC-5. Likewise, the overall improvement over the baseline algorithm is smaller. This may indicate that the parameters were over-trained for TREC-5. It may also merely indicate that the topics in TREC-6 are harder than the topics in TREC-5.

As mentioned in Section 2.1, we are concerned mostly with recall at 1000 documents and precision at 30 documents. In reality we consider the latter statistic to be the single most important, since in our view users are likely to look at only the first two screens of search results for any query. These are the statistics we report in Table 2, for TREC-6, and Table 3, for TREC-5. We show the statistics on a per-query basis, to illustrate that many of these methods work only for queries of a certain structure, or those with many — or few — relevant documents. For comparison with other TREC researchers, we also show the averaged uninterpolated average precision for each algorithm.

In Figure 1 we show the recall-precision curves for TREC-6 and TREC-5.

[3]This is not technically true, since the 0.3 score cutoff still applies: we look at less than 20 documents if not all 20 have score above 0.3 after the original retrieval step.

		base		+phrase		+syntax		+idf		+len		+feedback		+cluster	
Q#	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel
301	474	0.0333	60	0.0667	45	0.0333	44	0.1000	81	0.3333	94	0.4000	125	0.2333	77
302	77	0.2667	27	0.3333	39	0.3333	39	0.5667	68	0.6000	65	0.6000	71	0.6333	70
303	10	0.0333	9	0.1667	10	0.1667	10	0.1667	10	0.2000	10	0.1667	10	0.1333	10
304	226	0.0333	34	0.0667	63	0.0667	66	0.2333	86	0.4000	88	0.3667	94	0.2667	93
305	35	0.0000	3	0.0000	10	0.0000	10	0.0000	13	0.0000	9	0.0000	6	0.0000	10
306	352	0.2667	75	0.3000	139	0.3000	139	0.2333	122	0.4333	126	0.5333	163	0.4667	145
307	215	0.0000	24	0.0333	22	0.0333	22	0.0667	42	0.2667	50	0.3000	69	0.3000	69
308	4	0.0000	2	0.0667	3	0.0667	3	0.1000	4	0.1000	4	0.1000	4	0.1000	4
309	3	0.0000	1	0.0000	0	0.0000	0	0.0000	0	0.0000	0	0.0000	2	0.0000	2
310	13	0.0333	2	0.0333	3	0.0333	3	0.0667	6	0.0667	6	0.1333	6	0.1333	6
311	186	0.0000	6	0.0000	7	0.0000	9	0.0000	8	0.0333	10	0.0333	10	0.0333	9
312	11	0.0000	1	0.0000	1	0.0333	6	0.0000	5	0.0000	6	0.0000	7	0.0000	7
313	107	0.7333	68	0.7667	69	0.7667	69	0.8000	70	0.9000	59	0.9333	73	0.9333	73
314	45	0.0333	13	0.2333	11	0.2333	11	0.1333	34	0.1000	33	0.1333	27	0.1333	27
315	67	0.0000	0	0.0000	1	0.0000	1	0.0667	29	0.0667	31	0.0667	30	0.0667	29
316	35	0.0667	6	0.0000	4	0.0000	4	0.1000	12	0.1000	12	0.1000	15	0.1000	12
317	14	0.1000	8	0.0333	3	0.0333	3	0.3333	11	0.3333	11	0.3333	10	0.3333	10
318	128	0.0000	15	0.0000	15	0.0000	15	0.0000	21	0.1333	25	0.1000	22	0.0667	20
319	187	0.2000	54	0.1000	49	0.1000	49	0.2333	78	0.2000	68	0.3333	91	0.2667	85
320	6	0.0000	5	0.0000	6	0.0000	6	0.0333	6	0.1333	6	0.0333	6	0.0333	6
321	234	0.0667	29	0.1333	32	0.1333	40	0.1333	36	0.1667	20	0.2333	27	0.1000	21
322	34	0.0333	4	0.0000	7	0.0667	12	0.0333	19	0.0333	19	0.0333	16	0.0333	16
323	63	0.0333	9	0.3333	19	0.3333	19	0.1333	13	0.1000	12	0.1000	12	0.1000	12
324	162	0.2333	87	0.2333	89	0.2333	89	0.2333	102	0.2000	104	0.8333	125	0.8333	125
325	24	0.0667	11	0.0667	11	0.1333	13	0.2333	20	0.2333	20	0.2333	20	0.2333	20
326	48	0.0333	9	0.0333	1	0.0333	1	0.7333	44	0.7667	43	0.8000	46	0.8000	46
327	18	0.0000	6	0.0333	4	0.0333	6	0.0333	14	0.0667	12	0.1000	12	0.1000	12
328	9	0.0333	5	0.0667	7	0.1333	8	0.2333	8	0.2333	8	0.2667	8	0.2667	8
329	50	0.1333	15	0.1333	11	0.1333	11	0.1000	23	0.0667	21	0.0667	20	0.0667	20
330	60	0.0333	4	0.0000	15	0.0000	15	0.0000	7	0.0333	8	0.0333	19	0.0333	15
331	222	0.1667	24	0.3333	35	0.3333	35	0.5667	96	0.6000	119	0.6333	153	0.6333	146
332	278	0.1333	51	0.1667	56	0.1667	56	0.3000	89	0.4667	85	0.3667	87	0.3000	85
333	72	0.0667	18	0.0333	17	0.1333	21	0.2000	62	0.3333	61	0.4333	63	0.3333	62
334	18	0.0000	9	0.0333	11	0.1333	12	0.3667	18	0.4000	18	0.4000	18	0.4000	18
335	70	0.0333	13	0.1000	25	0.1667	26	0.3333	38	0.5333	38	0.6000	47	0.6000	47
336	12	0.0000	1	0.0000	0	0.0000	0	0.0000	3	0.0333	3	0.0333	3	0.0333	3
337	98	0.2000	16	0.3000	23	0.3000	23	0.3667	65	0.3667	63	0.3667	89	0.3667	89
338	5	0.0333	2	0.0333	2	0.0333	2	0.0333	2	0.0333	2	0.0333	5	0.0333	2
339	10	0.0333	3	0.0667	10	0.0667	10	0.1333	10	0.1333	10	0.1667	10	0.1667	10
340	81	0.1000	9	0.1000	22	0.1333	22	0.5333	38	0.5333	38	0.5667	38	0.5667	35
341	81	0.0333	13	0.1667	26	0.1667	26	0.2333	31	0.4000	34	0.4333	36	0.4333	38
342	23	0.0000	2	0.0000	2	0.0000	2	0.0000	4	0.0000	4	0.0000	5	0.0333	4
343	290	0.1000	13	0.1000	26	0.1000	40	0.1333	98	0.2667	107	0.4333	151	0.4333	155
344	5	0.0000	0	0.0000	1	0.0000	1	0.0000	4	0.0333	4	0.0333	4	0.0333	4
345	39	0.0333	3	0.0333	3	0.0333	3	0.0333	9	0.1667	14	0.1667	17	0.1667	17
346	106	0.0000	1	0.0000	2	0.0000	2	0.0000	7	0.0000	12	0.0000	13	0.0000	12
347	157	0.1000	14	0.1000	32	0.1000	32	0.1000	62	0.1333	78	0.0667	73	0.0667	73
348	5	0.0000	1	0.0333	2	0.0333	2	0.0333	4	0.0667	4	0.0333	4	0.0333	4
349	73	0.0000	5	0.0000	5	0.0000	6	0.0000	9	0.0333	13	0.0000	11	0.0000	12
350	69	0.0000	7	0.1000	15	0.1000	15	0.1667	16	0.2000	19	0.2333	25	0.2333	28
ALL	4611	0.0700	797	0.0987	1011	0.1087	1059	0.1727	1657	0.2207	1706	0.2473	1998	0.2333	1903
avg prec		0.0253		0.0459		0.0558		0.1292		0.1494		0.1703		0.1750	

Table 2: Per-query statistics for TREC-6. For each query we list the total number of relevant documents. For each algorithm, we record the precision at 30 documents and the recall at 1000 documents. We also report the average uninterpolated precision, averaged over all queries.

		base		+phrase		+syntax		+idf		+len		+feedback		+cluster	
Q#	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel	prec30	rel
251	579	0.3667	78	0.4333	65	0.1000	70	0.0667	69	0.0000	49	0.0000	31	0.0000	38
252	37	0.1000	9	0.1000	9	0.1000	9	0.1000	10	0.0000	8	0.0000	7	0.0333	12
253	10	0.0000	0	0.0000	0	0.0000	0	0.0667	2	0.0667	2	0.0667	6	0.0667	6
254	85	0.0000	1	0.0667	13	0.0667	23	0.0667	35	0.0667	36	0.1000	37	0.1333	41
255	109	0.0333	10	0.0333	4	0.0333	4	0.0000	3	0.0333	3	0.0000	3	0.0000	3
256	22	0.0000	3	0.0000	7	0.0000	7	0.0000	5	0.0667	11	0.0667	13	0.0667	13
257	135	0.1000	16	0.4000	33	0.4000	33	0.6000	76	0.6000	74	0.6000	92	0.6000	93
258	115	0.0000	11	0.0000	10	0.0333	21	0.1333	35	0.2000	37	0.3667	43	0.5000	53
259	36	0.0333	18	0.3333	28	0.4667	30	0.4333	33	0.4667	33	0.5000	35	0.5000	35
260	22	0.0000	4	0.0000	5	0.0000	10	0.0000	4	0.0000	2	0.0000	2	0.0000	2
261	87	0.1667	35	0.1667	53	0.1667	53	0.2000	54	0.3000	35	0.3333	35	0.3333	35
262	4	0.0667	4	0.1333	4	0.1333	4	0.1333	4	0.1333	4	0.1333	4	0.1333	4
263	15	0.0667	7	0.0000	11	0.0000	11	0.0000	11	0.0000	8	0.0000	8	0.0000	8
264	281	0.0333	14	0.0000	26	0.1000	48	0.3000	57	0.4000	60	0.5667	92	0.7333	118
265	147	0.0667	16	0.1667	31	0.2667	121	0.5000	127	0.7000	125	1.0000	128	1.0000	128
266	139	0.0000	2	0.0000	3	0.0000	3	0.0333	34	0.0333	32	0.0667	41	0.0667	41
267	4	0.0000	0	0.0000	0	0.0000	0	0.0000	1	0.0000	1	0.0000	1	0.0000	1
268	45	0.0000	8	0.0000	7	0.0000	4	0.0000	10	0.0000	14	0.0000	16	0.0000	16
269	594	0.1667	68	0.2000	66	0.2667	63	0.4667	80	0.4667	54	0.4333	78	0.4333	81
270	116	0.1333	43	0.3667	45	0.3667	45	0.5667	62	0.7667	59	0.9000	85	0.9000	85
271	86	0.0333	7	0.0667	12	0.0667	12	0.1333	24	0.1667	26	0.2000	30	0.2000	30
272	36	0.0333	14	0.2000	18	0.2000	18	0.4000	30	0.3667	28	0.3667	32	0.3667	32
273	513	0.2000	31	0.2000	37	0.3667	82	0.4333	113	0.5000	136	0.6667	298	0.7000	299
274	119	0.0000	38	0.1667	54	0.1667	54	0.3333	56	0.4000	50	0.7000	62	0.7000	62
275	19	0.0000	3	0.0000	0	0.0000	0	0.0000	5	0.1333	13	0.0667	15	0.0667	15
276	7	0.0667	7	0.2333	7	0.2333	7	0.2333	7	0.2333	7	0.2333	7	0.2333	7
277	74	0.0333	18	0.4667	45	0.4667	45	0.6667	51	0.6000	48	0.5667	51	0.5667	51
278	7	0.0000	1	0.0000	0	0.0000	2	0.0000	1	0.0000	1	0.0333	1	0.0333	1
279	2	0.0000	0	0.0000	1	0.0667	2	0.0000	2	0.0333	2	0.0333	2	0.0333	2
280	32	0.3000	21	0.4000	22	0.4000	22	0.3667	28	0.3667	25	0.4667	26	0.4667	26
281	1	0.0000	0	0.0000	0	0.0000	0	0.0000	1	0.0000	1	0.0000	1	0.0000	1
282	131	0.2000	36	0.0333	24	0.0333	24	0.2333	27	0.1333	28	0.2667	50	0.2667	50
283	84	0.0667	23	0.1333	14	0.1333	14	0.1333	18	0.0667	18	0.1667	44	0.1333	42
284	70	0.0000	4	0.0000	3	0.0000	6	0.1000	22	0.2333	28	0.3667	36	0.3667	37
285	261	0.4000	43	0.4000	52	0.4333	59	0.5000	143	0.5333	134	0.4667	139	0.4667	139
286	142	0.1000	65	0.7000	80	0.7000	80	0.7000	80	0.1667	16	0.2000	22	0.0333	22
287	40	0.1333	22	0.2333	26	0.2333	26	0.2000	28	0.2333	24	0.2333	25	0.2333	25
288	92	0.0000	10	0.1667	16	0.3667	27	0.4333	48	0.4667	53	0.4667	55	0.5000	55
289	141	0.0667	32	0.1333	32	0.1667	19	0.2333	58	0.3000	75	0.3000	73	0.2333	73
290	119	0.0000	14	0.0333	8	0.0333	8	0.0000	17	0.0667	29	0.0000	23	0.0000	23
291	407	0.0000	28	0.1667	24	0.1667	24	0.1000	13	0.1000	23	0.0667	16	0.0667	16
292	59	0.0000	3	0.0333	3	0.0333	4	0.0000	8	0.0333	5	0.0333	12	0.0000	3
293	41	0.0000	8	0.0000	4	0.0000	4	0.0667	8	0.0667	5	0.0667	5	0.0667	5
294	160	0.0000	9	0.0000	11	0.0000	13	0.0000	19	0.0000	23	0.0333	25	0.0000	26
295	15	0.1667	6	0.1333	6	0.1333	6	0.1667	9	0.1667	8	0.1667	8	0.1667	8
296	1	0.0000	0	0.0000	0	0.0000	0	0.0000	0	0.0000	0	0.0000	0	0.0000	0
297	86	0.0000	1	0.0333	19	0.0333	19	0.6333	23	0.6333	27	0.7333	39	0.7667	39
298	91	0.1333	13	0.1667	13	0.1667	13	0.2667	14	0.1667	26	0.3333	50	0.3000	49
299	62	0.1000	8	0.1333	10	0.1333	10	0.0667	11	0.0333	9	0.0333	11	0.0333	11
300	44	0.0000	1	0.0000	2	0.0000	2	0.0667	4	0.0667	18	0.0667	24	0.0667	24
ALL	5524	0.0673	813	0.1327	963	0.1447	1161	0.2027	1580	0.2113	1533	0.2493	1939	0.2513	1986
avg prec		0.0216		0.0618		0.0761		0.1192		0.1216		0.1489		0.1535	

Table 3: Per-query statistics for TREC-5. We used this data to tune the paramcters for the algorithms we studied.

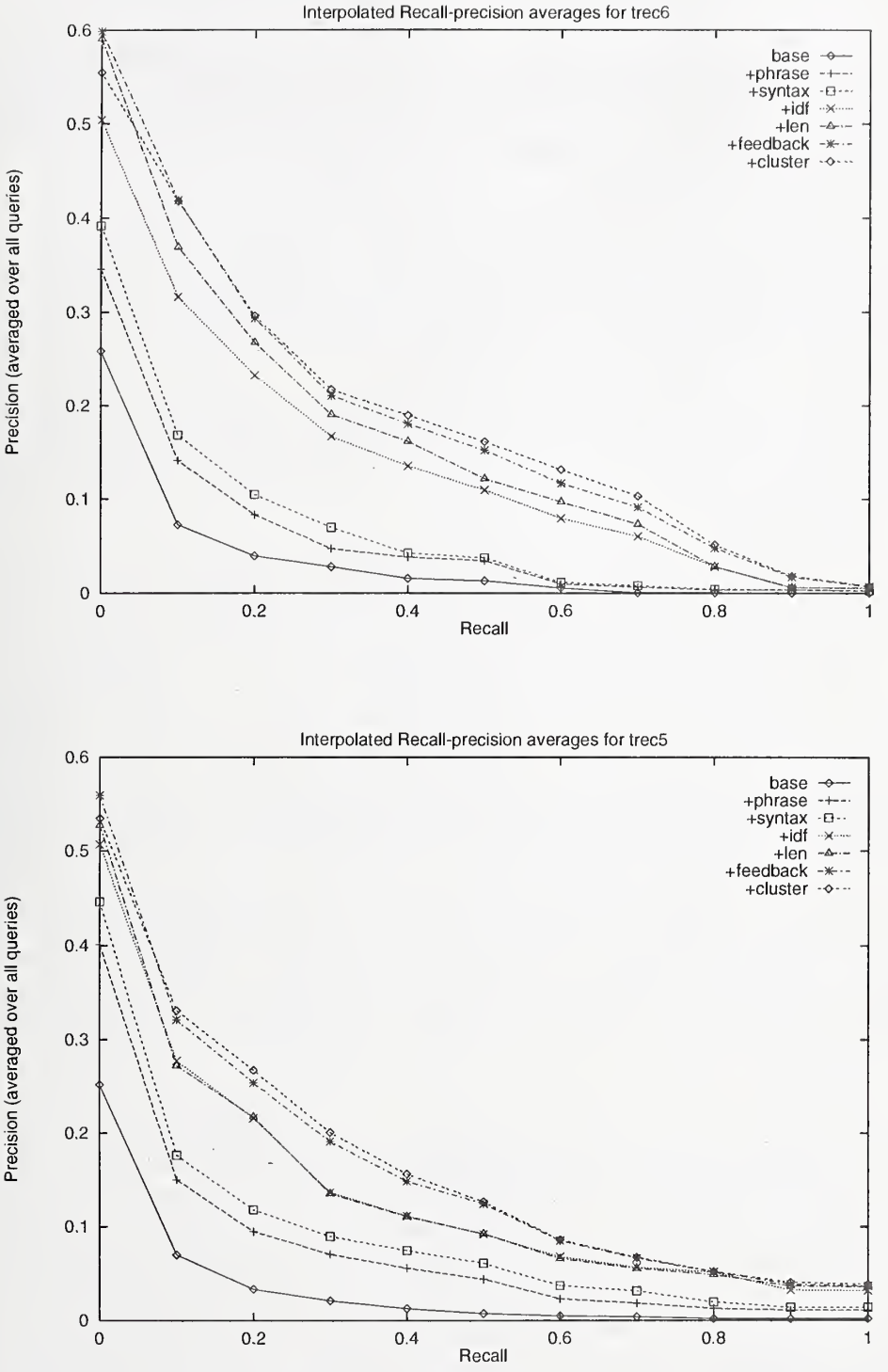


Figure 1: The interpolated recall-precision averages for the six algorithms discussed in this paper. The numbers are averaged over all 50 topics in TREC-6 (top figure) and TREC-5 (bottom figure).

3 The Routing Task

3.1 Method

Unlike our *ad hoc* submission, our *routing* submission focused on only using built-in functionality of the Verity search engine[4]. Since the routing task has training data in the form of positive and negative examples (documents), it made sense to use the built-in Verity Query By Example (QBE) functionality, which is manifested in Verity Query Language (VQL) using the `<LIKE>` operator. This operator takes several parameters:

- any number of positive examples,
- any number of negative examples, and
- the number of terms to include in the final query.

From these parameters, it then generates a query in VQL using only term frequency to perform term selection. It is important to remember that VQL does *not* have built-in support for IDF or document length normalization. Thus, compared to other TREC entries, the Verity QBE is at something of a disadvantage.

To train our system, we generated a classifier (a QBE query) for each routing query. The training process had to select which positive and negative examples to use, along with the number of query terms. Because of the large numbers of positive and negative examples, investigating all subsets of available examples was not tractable. Instead, we used a variation of "local region" training [Schütze et al., 1995], wherein training examples were first ranked using a QBE query with the actual TREC topic text as the single (positive) example. Selection of both positive and negative training examples was then based on the number of top ranked examples. Thus, training a classifier became a matter of selecting the number of top ranked positive examples, the number of top ranked negative examples, and the number of terms. Note that, except for determining the ranking of training examples, the original query text is not used.

We used average precision as a performance metric during training, since preliminary experiments indicated that it was more predictive of test set performance than precision at 30 documents (our preferred measure). Our preliminary experiments also indicated that as a function of the classifier parameters, average precision presented quite a "bumpy" optimization surface. Thus, more sophisticated optimization procedures did not seem helpful, and a version of grid-searching was used to select the parameters used for each query. Table 4 shows what values of parameters were used during training.

Two training regimens were used in conjunction with the grid-search. Both used only documents from "similar" sources as the test documents. The first regimen (regimen A) used training documents from the AP, WSJ, SJM, and FBIS corpora, with the best query parameters selected based on performance on all four corpora combined. These corpora were chosen because they seemed most similar in content and style to the test collection (more FBIS documents). Regimen B used a more aggressive source-sensitive approach. Here, training documents from AP, WSJ, SJM, and FBIS were used again, but all FBIS training examples were artificially ranked above documents from other collections. Furthermore, query parameters were selected based on performance on FBIS alone. Thus, regimen B was essentially just training on FBIS documents, but allowed the use of documents from the other collections when larger numbers of training examples were used.

[4]One slight change to the basic query construction procedure was made which theoretically should not significantly change performance, but which did improve the correlation between training and test set performance and also ameliorated the saturation problem. This change has since been incorporated into the next release of Verity's engine.

Parameter	Values Used for Training
# Positive Examples	5 10 30 80 160 all
# Negative Examples	0 5 10 20 40 160
# Terms	3 8 10 12 15 30

Table 4: Parameter Values Scanned to Train each QBE Query

Validation experiments on a portion of the training FBIS collection indicated that each regimen worked better for some queries than for others. Consequently, we decided to use a system selection approach, choosing a regimen individually for each query. Time constraints prevented investigation of different selection criteria, so we simply used whichever training regimen performed better in our validation runs.

In summary, our routing approach investigated several techniques based on using QBE straight "out-of-the-box":

1. training on a per-query basis,
2. local region training,
3. source specific training, and
4. system selection.

3.2 Results

A precision/recall graph of our submission (VrtyRT6) which used the system selection approach is shown in Figure 2. Also shown are graphs for both regimens and for an approach (labeled "fixed") which did not tune parameters on a per-query basis, but instead always used all positive examples, no negative, and 15 terms.

The system selection procedure is superior to regimen A but inferior to regimen B. Apparently our selection rule was not a good one, and did slightly worse than randomly choosing one of the two regimens. Our validation experiments showed the same behavior, but in those runs both regimens performed at about the same level, so the combined system showed neither improvement nor degradation. These results should not be surprising given that previous work on system selection has generally failed [Diamond, 1996].

Setting aside the system selection run, we note that both regimens perform better than the "fixed" run. Thus, it pays to tune the number of terms, the number of top positive examples and the number of top negative examples on a per-query basis. Furthermore, since regimen B does better than A, it also appears that it pays to emphasize training documents from the same source as the test set. Also, since about half of the trained queries resulting from regimen B made use of documents outside of the FBIS collection, it also seems important to make these extra documents available, instead of training exclusively on same-source documents. Our validation runs also support this conclusion. When the FBIS training data was split in half, with one half used for training and one for testing, regimen B consistently did better than only training on the FBIS documents (0.24 average precision versus 0.17-0.22, depending on which half was used for training). Finally, it appears that some queries were able to take advantage of the sorted order of training examples (the "local region" technique). About 33 of the 47 regimen A queries used less than the full number of positive examples.

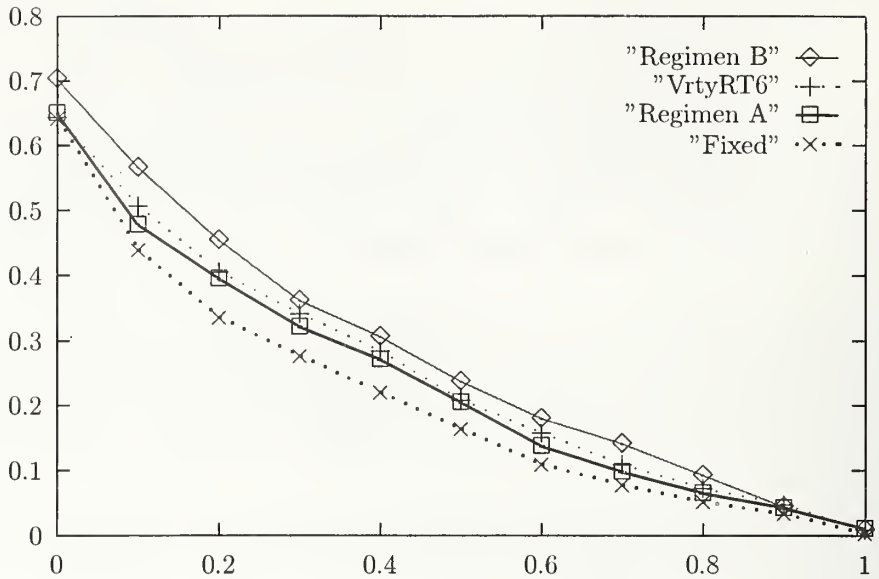


Figure 2: Precision vs. Recall for Routing Runs

Query	# Pos	# Neg	# Terms
3	all (561)	5	10
44	30	160	3
114	30	10	10
148	all (33)	40	3

Table 5: Parameters for Four Best Routing Queries

Figure 3 shows how the regimen B run compared to other routing entries. Overall, the performance is about average. However, only 16 queries exhibited above-median performance. Four were very significantly above the median, none were maximum, and only one was the worst. Over all queries, the mean average precision was approximately the same as the mean median score of all TREC entries.

Table 5 shows the parameters for the four best queries. No discernible pattern emerges, which supports the hypothesis that the parameter settings vary greatly according to the query.

Perhaps the most striking thing to note is that despite a lack of length normalization and IDF, our system was capable of producing reasonable results, on a par with the average TREC system. Judging by the boosts these techniques gave our *ad hoc* entry, it would not be unreasonable to assume the Verity's *routing* entry may have easily been more competitive.

4 Conclusions

Our results in the *routing* task indicate that despite a lack of standard techniques like IDF and length normalization, Verity QBE can perform on a par with other TREC entries, with appropriate selection of training examples and query length. Specifically, training on positive and negative examples which are "closest" to the original query text (the local region technique) provides a boost. Also, emphasizing training documents from the same source as the test collection helps.

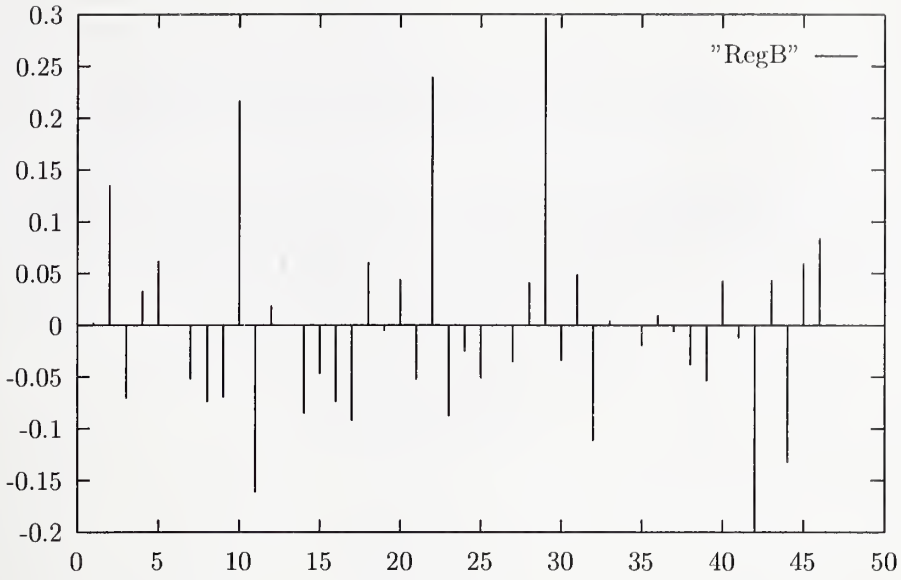


Figure 3: Routing Comparison to Median by Query for Regimen B

Both of these techniques presumably work by improving both term selection and term weighting. Also, by training a single classifier per query, and adjusting the number of terms in the resultant query, we were able to more accurately capitalize on the idiosyncrasies of each query.

The baseline *ad hoc* system is not nearly as impressive as the *routing*, presumably because it only has access to terms from the original TREC topic, and not from training examples. However, by selecting and weighting terms based on linguistic information and IDF, along with a novel length normalization approach and pseudo-feedback, we are able to significantly improve performance. By including these enhancements, we achieve performance on par with other TREC entries.

Most significantly, our experiments did not modify the well-established VQL extended boolean query language. As such, any of these enhancements could easily be incorporated as a built-in facility.

Acknowledgements

The authors would like to express special thanks to George Politowski who prepared the data, built Verity collections, and assisted the authors with the Verity toolkit (VDK).

References

[Diamond, 1996] Diamond, T. (1996). Information retrieval using dynamic evidence combination. PhD Dissertation Proposal. http://www.nwlink.com/~tgdiamon.

[Schütze et al., 1995] Schütze, H., Hull, D. A., and Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In Fox, E. A., Ingwersen, P., and Fidel, R., editors, *SIGIR 95: Proceedings of the Eighteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 229–237, Seattle. Xerox PARC, ACM Press.

ANU/ACSys TREC-6 Experiments

David Hawking, Paul Thistlewaite and Nick Craswell
Co-operative Research Centre For Advanced Computational Systems
Department Of Computer Science
Australian National University
{dave,pbt,nick}@cs.anu.edu.au *

January 10, 1998

Abstract

A number of experiments conducted within the framework of the TREC-6 conference and using a completely re-engineered version of the PArallel Document Retrieval Engine (PADRE97) are reported. Passage-based pseudo relevance feedback combined with a variant of City University's Okapi BM25 scoring function achieved best average precision, best recall and best precision@20 in the Long-topic Automatic Adhoc category. The same basic method was used as the basis for successful submissions in the Manual Adhoc, Filtering and VLC tasks. A new BM25-based method of scoring concept intersections was shown to produce a small but significant gain in precision on the Manual Adhoc task while the relevance feedback scheme produced a significant improvement in recall for all of the Adhoc query sets to which it was applied.

1 Introduction

The work reported here comprises a number of text retrieval experiments conducted within the framework of TREC-6 and addressing questions of interest in the following research areas: Scalable information retrieval; Relevance Feedback; Distance-based relevance scoring; Selective Dissemination of Information and Automatic Query Generation. ANU/ACSys completed Automatic and Manual Adhoc, Filtering and VLC tasks.

1.1 Relevance Scoring Methods Employed

Three different methods of relevance scoring were employed in the experiments reported here:

Frequency: Documents are scored using the Cornell variant of the Okapi BM25 weighting function [Singhal et al. 1995; Robertson et al. 1994].

$$w_t = tf_d \times \frac{\log(\frac{N-n+0.5}{n+0.5})}{2 \times (0.25 + 0.75 \times \frac{dl}{avdl}) + tf_d}$$

where w_t is the relevance weight assigned to a document due to term t, tf_d is the number of times t occurs in the document, N is the total number of documents, n is the number of documents

*The authors wish to acknowledge that this work was carried out within the Cooperative Research Centre for Advanced Computational Systems established under the Australian Government's Cooperative Research Centres Program.

containing at least one occurrence of t, dl is the length of the document and $avdl$ is the average document length.

Concept: Groups of related terms in a query are called concepts. Documents are scored against each concept and the results are recorded in separate accumulators. The final score s for a document is derived from the concept scores $c_1, \ldots c_n$ using $s = (kc_1 + 1) \times \ldots \times (kc_n + 1)$. In frequency scoring, a document with many occurrences of only one concept may score more highly than another which contains evidence for all the concepts. Concept scoring is designed to boost the weight of documents with evidence for the presence of all concepts.

Distance: Documents are scored using the lexical-distance between instances of concept members as described in [Hawking and Thistlewaite 1996; Hawking et al. 1996]. This method does not require collection frequency statistics.

1.2 Hardware and Software Employed

Since TREC-5, the PArallel Document Retrieval Engine (PADRE) has been completely rewritten to operate on workstations and clusters of workstations. The new PADRE97 software [Hawking 1997b] was used in all experiments reported here. A single-processor Sun Ultra-1 was used except in runs for the VLC track, where a cluster of DEC Alphas was employed.

Interactive query modification was carried out using a new graphical user interface to PADRE97 (`quokka`) which has been designed to facilitate the construction of queries suitable for Concept and Distance as well as Frequency scoring.

1.3 Statistical Testing of Differences Between Runs

Throughout this paper, wherever comparisons are made between pairs of runs, apparent differences between means have been tested for statistical significance using two-tailed t-tests[1] with $\alpha = 0.05$.

2 Automatic Query Generation

Automatic AdHoc, Official Runs `anu6alo1` and `anu6ash1`, semi-official run `anu6avs2` and various unofficial runs.

The goal of experiments using automatic query generation was to provide preliminary answers to the following questions:

1. Using the Frequency scoring method defined above, can the performance of queries be improved by the addition of pseudo-phrases automatically extracted from the query?
2. Using the Frequency scoring method, what is the optimum method for finding and using additional pseudo relevance feedback terms?

Automatic runs were performed for all three official sub-categories: full, description-only, and title-only. In addition, runs were performed using queries derived from title-plus-description. The basic strategy in each case was:

1. generate stems and two-stem phrases from the allowable parts of the topic descriptions;

[1] Future consideration will be given to following the advice of Savoy [1997] who recommended the use of medians rather than means and the use of statistical bootstrapping techniques.

2. score documents against the resulting query; and

3. optionally, update document scores using the additional terms suggested by pseudo relevance feedback.

2.1 Phrases

In PADRE, phrases within documents are identified by computing a `followed-by` proximity relation between the matchsets for all the terms in the phrase.

In generating query phrases, the allowable text of each topic description was converted into a sequence of stemmed non-stopwords and phrase-end markers. A phrase-end marker # was inserted for each SGML tag, for each punctuation mark (except hyphens not surrounded by spaces), for each stopword, and at the end of the topic.

In such token sequences, each contiguous (ordered) pair of stems was considered to be a phrase. Thus, the token sequence `# A B C # D #` would generate the phrases `"A B"` and `"B C"` only. Would-be phrases interrupted only by one of `in`, `to`, `of`, `for`, `on`, or `with` were also accepted and the phrase proximity parameter was increased accordingly when processing documents.

2.2 Relevance Feedback

Subsequent references to *relevance feedback* in fact refer to *pseudo relevance feedback* as there was no human involvement in the feedback process. Instead, highly ranked documents retrieved by an initial query were assumed to be sufficiently relevant as to constitute a useful source of additional query terms.

Robertson [1990] argued that the weights used to select terms to be added to a query should, in general, be different from the document term weights used when processing the query. This approach has been taken here.

2.2.1 Method of Term Selection

Instead of mining complete document text for new terms, only the *hotspots* were mined. A hotspot was defined as a contiguous passage of text within a document which lies within a specified p characters of a term or phrase occurrence. All the hotspots within the T top-ranked documents resulting from running the initial query were mined for new terms. Stopwords and terms from the initial query were not considered. All other terms were stemmed and stored in a hash table and their frequencies of occurrence within the hotspots were accumulated.

Once all hotspots had been mined, selection values for each term in the hash table were computed according to the formula given by Robertson [1990]:

$$a_t = w_t(p_t - q_t)$$

In Robertson's work the p_t and q_t were the probabilities that a relevant and a non-relevant document, respectively, contained the term t. Here, p_t and q_t are the probabilities that any particular term in a hotspot and not in a hotspot, respectively, is the specific term t. That is,

$$p_t = tf_h/l_h$$

and

$$q_t = (tf_C - tf_h)/(l_C - l_h)$$

where tf_h is the frequency of the term in the hotspots, tf_C is the frequency of the term in the whole collection, and l_h and l_C are the number of words in the hotspots and in the complete collection respectively. Robertson's w_t was the relevance weight of the document due to term t but here an approximation was used because hotspots rather than whole documents were examined. Furthermore, in the PADRE97 version employed in the experiments, document frequencies were not stored in the term dictionaries and were thus relatively expensive to compute. Accordingly, the document frequency was estimated from the raw frequency by dividing the latter by 3. For the purpose of term selection, it was assumed that $dl = avdl$, and that $tf_d = 1$, allowing the document term weighting formula to be simplified to:

$$w_t = \frac{\log(\frac{N - tf_C/3 + 0.5}{tf_C/3 + 0.5})}{3}$$

2.2.2 Relevance Feedback Training

Relevance feedback in the above-described scheme is controlled by four parameters: T (the number of top-ranking documents to mine for feedback terms, p (the proximity limit defining the extent of the hotspots), n (the number of new terms to add), and w_0 the query term weight to be given to the best new term. A series of experiments over 50 TREC topics (to be described elsewhere) was used to pick a set of values for these parameters.

Table 1: Effectiveness of relevance feedback on past TREC Automatic AdHoc tasks using the feedback parameters ($T = 20; p = 500; n = 30; w_0 = 0.5$). All differences were statistically significant.

	No Feedback	Feedback		
Task	Ave_Prec	Ave_Prec	Precision @ 20	Recall
TREC-3 short	.2063	.3018 (+46%)	+28%	+15%
TREC-4 short	.1925	.2498 (+30%)	+17%	+14%
TREC-5 short	.1502	.1959 (+30%)	+21%	+17%
TREC-3 long	.3441	.3748 (+9%)	+7%	+3%
TREC-5 long	.2356	.2515 (+7%)	+5%	+4%

To confirm the generality of the chosen values, training runs using these parameters were performed on the TREC-3 (short and long), TREC-4, and TREC-5 (short and long) tasks. Results obtained are shown in table 1. In every case, on all three measures, there was a statistically significant benefit from using relevance feedback.

The gain was much smaller for the long topics than for the short. This may seem counter-intuitive, as one might expect that better initial queries would yield better text from which to mine relevance feedback terms. However, it is possible that queries derived from the long topic descriptions are closer to optimal, restricting the scope of potential gains from relevance feedback.

Another possibility is that the benefit of relevance feedback terms was scaled down because of higher term frequencies in the longer query. This possibility has yet to be investigated.

2.2.3 Relevance Feedback Failures on Training Topics

The relevance feedback scheme adopted above ($T = 20; p = 500; n = 30; w_0 = 0.5$) produced quite consistent improvement in training. Considering the short topic tasks, on only 23 of the

149 topics did relevance feedback result in loss of more than 0.005 in average precision. Only 5 of the 149 topics were harmed by more than 0.05 and only one by more than 0.10. This topic however saw a loss of 0.47 in average precision! In this case, the unexpanded query achieved an average precision of 0.95. Consequently, additional terms were almost guaranteed to reduce performance.

In earlier experimentation, feedback failures were much more common and consideration was given to the design of a mechanism for turning off relevance feedback on queries which exceeded a threshold of estimated risk. No such mechanism was used in runs reported here.

2.2.4 Parameters Used in Official TREC-6 Runs

Final runs in the very-short, short and long automatic adhoc category were all performed with $(T = 20; p = 500; n = 30; w_0 = 0.75)$ as there was some evidence that a slightly higher value of w_0 might perform better.

2.3 Automatic Adhoc Results

Results for Automatic Adhoc runs are summarised in Tables 2, 3 and 4 and plotted in Figure 1.

Relative to all 57 Category A Automatic Adhoc runs, long-topic run `anu6alo1` retrieved more relevant documents than any other run and achieved best overall precision@20 results. Only the City University title-only run achieved better overall average precision.

The method used in run `anu6alo1` did not perform as well when applied to the description-only and title-only tasks either in absolute terms or relative to all other official submissions in those categories. In these tasks, relative performance was better on the recall rather than on the precision dimension. For example, the unofficial title-only run would have ranked second (of 12) on recall (percentage of all relevant documents retrieved) but eighth on early and average precision.

Title vs. Description: As reported by other groups, the ANU/ACSys title-only run (`anu6avs2`) apparently out-performed the description-only run (`anu6ash1`) by a considerable margin (35% in average precision, 16% in precision @20 and 9% in recall). However, due to large variance in the results, none of these differences was statistically significant ($t(49) = 1.74, 1.21, 1.43$ respectively)!

Title plus Description: A run anu6atd1 using both title and description fields performed 31% better on average precision, 23% better on precision @20 and 16% better on recall than the description-only run. All these differences were statistically significant.

Value of phrases: When all phrases were removed, performance of the `anu6alo1`, `anu6ash1` and `anu6avs2` runs diminished by a small percentage on each of the three measures (average precision, precision@20 and recall). Only in the case of `anu6ash1` precision@20 was the difference statistically significant. Even combining the three runs failed to yield statistically significant differences.

Effectiveness of Relevance Feedback: The effect of relevance feedback in the TREC-6 task is reported in Table 5. As may be seen, feedback produced a statistically significant gain in recall for each of the query sets. The percentage gain in recall for the full-topic queries was similar to that achieved in training on the TREC-3 and TREC-5 long topics (see Table 1). For the short forms of the topic, however, feedback produced much smaller percentage gains in recall than were achieved in training. In contrast to the training results, the apparent improvements in average precision on TREC-6 were not statistically significant.

The possibility that the poorer performance of feedback may have been due to the use of $w_0 = 0.75$ rather than the value of 0.5 used in training was investigated post hoc by re-

Table 2: Average Precision performance of ANU/ACSys Automatic Adhoc runs relative to all official runs in the same category. The number of topics for which the run achieved best (possibly equal best) performance and the number achieving median or better are tabulated in the last two columns. The unofficial title-plus-description run is compared to the group of official description-only runs. There were 50 topics.

Run-id	Category	Mean	Rank	#best	#≥med.
anu6alo1	Full	.2602	1/16	10	49
anu6ash1	Desc. only	.1645	15/29	3	35
anu6avs2	Title only	.2216	8/12*	0	20
anu6atd1	Title/Desc.	.2157	NA	11	41

Table 3: The same runs as in table 2, compared on the basis of overall recall (percentage of all relevant documents retrieved). The figures in parentheses in the #best column show the number of topics for which all relevant documents were retrieved.

Run-id	Category	Percent	Rank	#best	#≥med.
anu6alo1	Full	62%	1/16	14(8)	48
anu6ash1	Desc. only	48%	8/29	11(7)	45
anu6avs2	Title only	55%	2/12	10(7)	30
anu6atd1	Title/Desc.	59%	NA	21(9)	46

Table 4: The same runs as in table 2, compared on the basis of overall recall (percentage of all relevant documents retrieved). Topic-by-topic precision@20 data was not available for all runs.

Run-id	Category	Mean	Rank
anu6alo1	Full	.379	1/16
anu6ash1	Desc. only	.282	8/29
anu6avs2	Title only	.327	8/12
anu6atd1	Title/Desc.	.348	NA

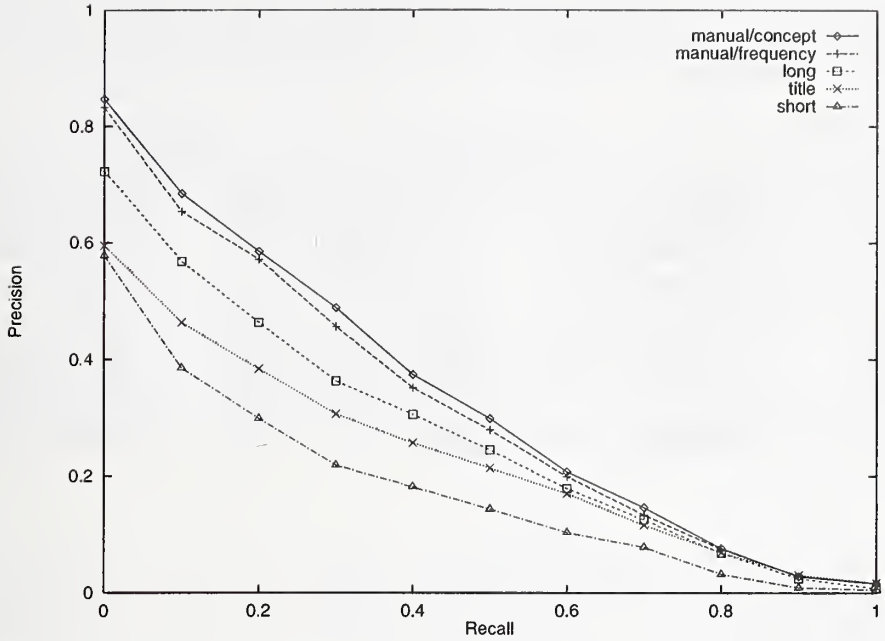


Figure 1: ANU/ACSys Adhoc runs. The two top lines correspond to the same interactively developed set of queries scored using two different methods.

running `anu6ash1` using the lower value of w_0. The new run (`anu6ash3`) achieved slightly better performance than the original (see Table 5) but apparent differences in average and early precision were still not significant. Compared to the training results reported in section 2.2.3, feedback harmed average precision results for a much higher percentage of topics (30% cf. 15% by more than 0.005, 8% cf. 3% by more than 0.05, and 8% cf. 1% by more than 0.10). If feedback could have been selectively switched off for the topics where it did harm, overall average precision would have risen to 0.1867, an increase of 20% over the non-feedback case. This increase is still markedly smaller than that observed in each of the three short-topic training sets, despite the inclusion of feedback failures in the latter results.

2.4 Automatic Adhoc Discussion and Conclusions

Title vs. Description: The text making up the description field of the topic statements appears to be designed to augment rather than to serve as an alternative to the title. For example, the title field of topic 312 contains the highly precise term *Hydroponics* whereas the description *Document will discuss the science of growing plants in water or some substance other than soil* gives an explanation using less precise terms. It is therefore not at all surprising that inclusion of the title terms improved results.

Value of phrases: Despite the lack of a statistically significant result, the use of query phrases as described has led to apparent improvements in nearly every training and official run in which the comparison has been made. There seems to be no reason to discontinue their use.

Value of relevance feedback: The relevance feedback scheme used delivered a consistent apparent benefit (averaged over 50 topics) on each of the tasks to which it was applied. However, its effectiveness on the TREC-6 task would have been much less than had been observed on tests with sets of earlier TREC tasks, even if feedback had been disabled on topics where it did

Table 5: Effectiveness of relevance feedback on TREC6 AdHoc tasks. Feedback parameters ($T = 20; p = 500; n = 30; w_0 = 0.75$) where used in all cases except for run `anu6ash3` where $w_0 = 0.5$. Statistically significant differences are marked with an asterisk.

		No Feedback	Feedback		
Run-id	Task	Ave_Prec	Ave_Prec	Prec@20	Recall
`anu6avs2`	Title-only	.2113	.2216 (+5%)	+11%	+7%
`anu6ash1`	Description-only	.1556	.1645 (+6%)	+9%	+9%*
`anu6ash3`	Description-only	.1556	.1680 (+8%)	+8%	+11%*
`anu6atd1`	Title plus desc.	.2035	.2157 (+6%)	+7%	+7%*
`anu6alo1`	Full topic	.2461	.2602 (+6%)	1%	+4%*
`anu6man1`	Manual (Blind)	.2668	.2785 (+4%)	+2%	+4%*
`anu6min1`	Manual (Interact.)	.3044	.3172 (+4%)	+1%	+3%*

harm.

Contrary to results on earlier TREC tasks, the size of the benefit was similar regardless of the length of the initial query. Table 5 shows that there was a consistent apparent benefit on the three measures (even for manually improved queries). The benefit to recall was statistically significant in six out of the seven cases considered.

The current feedback scheme seems sufficiently robust to justify its routine use, particularly where high recall is important. However, it is hoped that further refinement may result in an adaptive system which reduces harm and magnifies benefits.

3 Manual Query Generation

Manual AdHoc, Official Run `anu6min1`

3.1 Manual Query Generation Process

A relatively naive user generated a series of manual query sets by successively refining an initial automatically-generated set. In this way it was possible to compare *blind* (no interaction with documents) manual improvements with those obtained after interaction with the test documents. Details of the process were as follows:

Automatic queries were generated from the full topic descriptions using an earlier version of the automatic query generator described above. (The queries were similar but not identical to the queries used in `anu6alo1` without feedback.) The topics and queries were then presented to a relatively naive user of the `quokka` graphical user interface to PADRE. The user was asked to improve the initial queries using any of the following techniques:

1. remove any terms which appeared likely to be distractors,
2. combine any suitable pair of words into a phrase,
3. add new terms which were obviously missing,
4. alter query term weights.

Table 6: Summary of manual runs. The automatic long-topic run feedback `anu6alo1nf` is included as a baseline. None of the runs in this table used relevance feedback. Query processing times are one-observation-only elapsed times observed on a non-dedicated Sun Ultra-1.

Run	Scoring	Ave_Prec	Prec@20	Recall	Time per query (sec.)
`anu6alo1nf`	freq.	0.2461	0.376	2657/4611	20.1
`anu6man1`	freq.	0.2668	0.413	2834/4611	10.1
`anu6con1`	freq.	0.2723	0.427	2929/4611	9.6
`anu6con2`	concept	0.2813	0.438	2980/4611	27.4
`anu6dis1`	dist.	0.0188	0.054	909/4611	276.1
`anu6min1`	freq.	0.3044	0.467	3042/4611	10.4
`anu6min1con`	concept	0.3168	0.486	3099/4611	31.9

Before working on the TREC-6 task, the user was given a training run over the TREC-5 task during which he could compare precision-recall plots for each topic before and after his modifications.

This first phase of manual modification resulted in a set of queries which were used in unofficial run `anu6man1`.

Table 7: Comparison of frequency and concept scoring for two sets of manual queries. Each measure shown is the average of fifty individual topic results. Asterisks indicate statistical significance.

	Frequency Scoring			Concept Scoring		
Run	Ave_Prec	Prec@20	Recall	Ave_Prec	Prec@20	Recall
Blind	0.2723	0.427	0.7429	0.2813(+3%)*	0.438(+3%)*	0.7488(+1%)
Interact.	0.3044	0.467	0.7615	0.3168(+4%)*	0.486(+4%)*	0.7704(+1%)*

Next, the same user was asked to group the terms in the `anuman1` queries into *concepts*. The following explanation of concepts is similar to that given to the user.

> In judging relevance of documents to the topic, "What is the economic impact of recycling tyres?", you might decide that the topic involves three separate concepts: *economic impact*, *recycling* and *tyres* and that relevant documents are likely to contain evidence for the presence of each of them. There may be a whole list of words or phrases which could serve as evidence for the presence of a concept. For example, `profits`, `losses`, and `benefits` might constitute evidence for the presence of *economic impact*.

Sometimes, during this process, new terms suggested themselves. The resulting queries were used in unofficial runs `anu6con1`, `anucon2` and `anudis1`, corresponding to frequency, concept ($k = 30.0$) and distance scoring.

3.2 Concept and Distance Scoring

The present authours have been interested for some time in the idea that queries or sub-queries can be viewed as concept intersections. For a document to be relevant to a topic, there should

be evidence for the presence of all of the concepts, not just one. Naturally, the scoring methods must take into account the possibility that evidence actually present may be missed by the query.

ANU/ACSys manual queries in TREC-4 and TREC-5 were scored using the lexical length of concept spans [Hawking and Thistlewaite 1996], but it was recognised that span-based queries were harder to generate. In TREC-5, efforts were made to use automatic methods to augment manually generated distance queries. These efforts were moderately successful and could have been more so had they been improved interactively.

An unfortunate aspect of distance-based scoring is that errors in defining concepts, such as placing synonyms in different concepts, may dramatically alter the performance of the query. The present work proposes *Concept* scoring (defined in Section 1.1) as a method with the potential to gain benefit from concept intersections without the degree of risk associated with distance scoring. Using the method, each concept was scored using the Frequency function as an independent sub-query and the resulting scores combined in a way which boosted the overall scores of documents containing more of the concepts.

An effort was made to compare the benefits of this concept scoring compared to span-scoring. However, there was insufficient time to test the new PADRE97 implementation of spans prior to use and results obtained may have been affected by coding bugs.

3.3 Interactive Manual

In the final stage of manual query refinement, the same user was allowed to interactively modify the concept queries by running them and scanning the documents retrieved. Unfortunately, due to a misunderstanding, this interaction was done over CD2/CD5 rather than CD4/CD5. This resulted in a new set of queries (sometimes using negative query term weights) which were subsequently re-run over CD4/CD5 as official run `anumin1`.

3.4 Manual Adhoc Results

Table 6 summarises the manual adhoc runs. Blind manual tweaking (including organisation into concepts and use of concept scoring) of the initial queries not only produced statistically significant benefits in average precision (+14%), precision @20 (+16%) and recall (+7%) but also halved the running time of the queries. By comparison, automatic feedback applied to the initial queries gained less than the manual tweaking and took six times as long to run.

Concept scoring worked significantly better than Frequency scoring in both of the cases shown in Table 7. The benefit is most evident in the precision rather than the recall dimension.

Distance scoring worked very poorly as shown by the results for run `anu6dis1` in table 6.

Interactive modification of queries produced statistically significant benefits in average precision (+12%) and early precision (+9%) despite the interaction using incorrect data. The apparent improvement of 3% in recall was not significant.

3.5 Manual Adhoc Discussion and Conclusions

The fact that a non-expert user was able to substantially improve both the speed and the effectiveness of good automatic queries in a short time (even without interaction with the documents) indicates that there is considerable scope for improvement in the automatic query generation process.

In the future, consideration will be given to using automatic queries generated from topic titles only as the starting point for manual runs, particularly in time-limited situations such as

the High-Precision track. It has been noted that non-feedback automatic queries generated from only the topic title found an average of 5.9 relevant documents in the first 20 retrieved compared with 7.5 for the corresponding long-topic versions, but took an average of only 1.3 seconds to process, compared with 20.1 seconds.

It is notable that both concept scoring and relevance feedback are (independently) capable of significantly improving the performance of Okapi-scored interactively developed queries. The results for concept scoring are encouraging but further work is required to confirm generality and to hopefully improve the method.

At the time of writing, it had not been determined whether the disappointing results for Distance scoring arose from bugs in the code, from poor construction of the concept groups, from poor automatic generation of the `span` commands used in scoring or for some other reason. Further work is needed to investigate and hopefully to rectify the cause of the poor performance and to further compare the three alternative scoring methods.

4 Filtering Experiments

Filtering Track, Official Runs `anu6fltU1` and `anu6fltU2`

Filtering queries were generated from topic descriptions and training judgments, using the programs `topic-aqg` and `freq-aqg`, and applied to the training collection to derive relevance score *thresholds*. Queries and thresholds were then applied to the test collection to generate the TREC-6 submissions.

The `topic-aqg` program was used to extract terms and two word phrases from topic descriptions, using methods similar to those used in the Adhoc tasks. Terms were weighted more highly if they appeared more than once, if they were written in capitals and if they appeared in the topic title. The resulting terms and phrases were ordered by decreasing weight.

The `freq-aqg` program was used to extract terms and two word phrases from training documents. Each term and phrase from the training documents was ranked and weighted according to to $P_r - P_i + 1$ where P_r is the probability of it occuring in a relevant training document and P_i was the corresponding probability for irrelevant training documents.

To generate a filtering query, the best n terms/phrases were taken from the `topic-aqg` ranking and the best m terms/phrases from the `freq-aqg` ranking (duplicates were not removed), and term weights from these sources were scaled by a factor of w_n and w_m respectively.

To assist in finding optimal query generation parameters, the Generalized Reduced Gradient (GRG2)[2] nonlinear optimization algorithm was employed. The decision variables were n, m, w_n and w_m, and the objective function was average utility F_1 across all topics when run on the training collection. As n and m increased, the utility tended to gradually but uniformly increase, so large values of n and m were chosen (n=80 and m=80). It was also found that values of $w_m \approx 1.1904$ and $w_n \approx 0.7317$ would achieve greater utility than other choices of scaling factors (7.5% greater utility than for $w_m = 1$ and $w_n = 1$). These optimal training-collection values were used in test-collection query generation[3].

Documents were scored according to the Okapi variant used in Adhoc runs, but document frequencies and collection sizes were taken from the training collection rather than the test collection. Thresholds for each topic were set at the PADRE document score cutoff corresponding

[2]Developed by Leon Lasdon, University of Texas at Austin, and Allan Warren, Cleveland State University and implemented in Microsoft Excel 97.

[3]It would be interesting to optimise over the test judgments now that they are available and to compare the resulting parameter values with those actually used.

to maximum utility on the training collection. Test documents failing to reach the threshold score were rejected.

4.1 Filtering Results

Table 8: Summary of the performance of ANU/ACSys Filtering runs. There were 47 topics. *Num_ret* is the average number of documents returned by the run while *Num_rel* is the average total number of relevant documents. *Zeros* shows the number of topics for which the run returned zero documents and (in parentheses) the number of topics for which a group best result was achieved by the run while returning zero documents.

Run-id	Ut. Function	Utility	Rank	#best	#≥med.	Num_ret	Num_rel	Zeros
anu6fltU1	F1 (prec.)	12.97	6/17	7	36	35	146	12(3)
anu6fltU2	F2 (recall)	-57.55	5/17	2	37	89	146	3(1)

Table 8 summarises the performance of the ANU/ACSys filtering runs. Performance was quite pleasing. Only two groups achieved better results on the F2 measure and three on F1.

The number of documents returned by the ANU/ACSys runs was, on average, much less than the total number of relevant documents, even in the F2 case. Note that returning zero documents achieves an F1 score of 0.0 but an F2 score of -146 (on average).

4.2 Filtering Discussion and Conclusions

Future work on the same basic filtering approach is likely to investigate:

- optimisation over more inputs (stemming/nonstemming, phrases/notphrases, different weighting profiles etc.);
- use of stemming;
- separate optimisation for F1 and F2;
- use of $n > 80$ and $m > 80$; and
- different term ranking and term weighting strategies.

5 Experiments with a Larger Collection

Very Large Collection Track, Official Runs `anu6vlb1` and `anu6vlc1`

The main goal of research in this area was to design, implement and test a scalable retrieval architecture. In the recent re-design of PADRE, attention was paid to minimising communication, minimising synchronisation points and maximising use of communication buffering. The bulk of this work has been reported elsewhere [Hawking 1997b; Hawking 1997a].

Figure 2 predicts the scalability of PADRE97 query processing for three different hardware environments. It suggests that as data size grows, query processing speed on a workstation will deteriorate approximately linearly with data size until physical memory limits cause paging and consequent more rapid degradation. By contrast, if the number of workstations in a cluster is

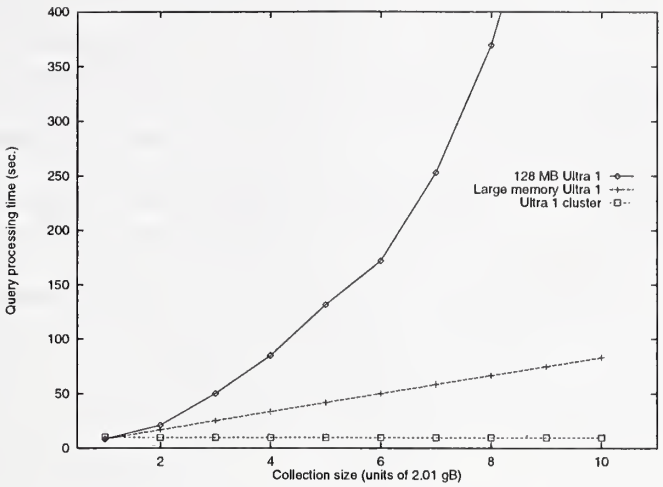


Figure 2: Elapsed query processing times for processing collection sizes measured in 2.01 gigabyte units on three different systems: Ultra 1 (observed times); Ultra 1 with memory hypothetically increased in proportion to the data size (projected times); a cluster of Ultra 1s with one search engine for each unit of collection size (scaled up from smaller data on SPARC-based Fujitsu parallel machine). (Reproduced from [Hawking 1997b].)

increased in proportion to the increase in data size and the data is evenly distributed across the cluster, then query processing times need not increase at all.

Hawking [1997b] also reported results for query processing over a 102 gigabyte (replicated) collection using ten 64-megabyte SGI workstations in a student laboratory.

Further work is needed to determine how scalability is affected by network latency when query processing times are short.

Unfortunately, the SGI laboratory was only available for a few days during student vacation and a VLC run conducted during that time was affected by a bug. Official runs in the VLC track were therefore processed on a cluster of eight DEC Alphas. Using eight nodes to process ten times as much data suggests that a VLC/baseline query processing time ratio of 1.25 would be an appropriate target. Achievement of this target assumes that times on each Alpha would increase by 25% in response to a 25% increase in data size and requires perfect load balance across nodes (unlikely to be achieved in practice).

Optimised queries were used (in which only the 15 lowest-frequency terms were processed) and only the top 20 documents were retrieved. Retrieving 1000 documents with the same queries increased the average time by 23% for the baseline. Query processing full (average 30 term) queries over the baseline retrieving 1000 documents took 217% longer. The initial queries used in both cases were generated from the full topic descriptions by an earlier version of the automatic query generator. They are thus similar to `anualo1nf`.

5.1 VLC Results

Table 9 shows the VLC measures taken from the two ANU runs in the VLC task. Results for all groups are presented in the VLC Track Overview elsewhere in these proceedings.

Of the seven official 20-gigabyte runs, the ANU/ACSys run:

1. required the least disk space (due to effective compression) despite full term-position in-

Table 9: Summary of ANU/ACSys VLC runs on a Alpha Farm consisting of eight 266 MHz EV5 Alphas connected by both 155 Mbit/sec ATM and switched 10BaseT. The only disk storage local to the Farm was a 20 gB RAID array connected by SCSI to one of the Alphas. Six of the Alphas (including the one supporting the RAID array) were equipped with 128 MB of RAM and the remaining two with 192 MB. Each Alpha features on-chip primary and secondary caches and an off-chip 2MB cache. The baseline run was carried out on the Alpha with directly connected RAID array. Indexing of the VLC was carried out sequentially using only the RAID-equipped node. The VLC query processing runs were carried out using 8 of the nodes. The RAID node ran the UIF process and also searched a small (1.1 gB collection). Disk space figures quoted exclude as-supplied compressed text (baseline: 0.80 gB vlc: 8.00 gB) but include all data structures generated, whether used in query processing or not. Costs were obtained from the Digital Equipment Corporation website on 15 September, 1997 and include the cost of the nodes and the RAID storage array. The VLC cost (only) includes the cost of the ATM switch connecting the nodes.

Measure	anu6vlb1 Baseline	anu6vlc1 VLC	VLC/Baseline
Precision@20	0.356	0.509	1.43
Ave. query processing time	12.1 sec.	50.5 sec.	4.17
Data struct. bld. time	1.405 hr.	15.6 hr.	11.1
Disk space	0.626 gB	6.06	9.68
Memory	128 MB	1152 MB	9
H/W cost (USD)	23.9	95.1	3.98
gB-queries/hour/kilo$	25.9	14.9	0.598

formation being included in indexes.

2. achieved second-best scalability of query processing time. However, the VLC/baseline ratio of 4.17 was much higher than the target (1.25).

3. achieved the third fastest indexing rate, despite using only a single processor. The two faster runs each used four or more processors.

4. achieved the third fastest query-processing rate. However, query processing was 20-40 times slower than the two faster runs.

5. ranked fourth on the bang-per-buck measure but was a factor of nearly 500 behind the best-scoring system.

Like all other groups, ANU/ACSys observed a large increase in early precision from the baseline to the VLC run. Comparison of actual precision values is not particularly meaningful because of a diversity of query construction methods used. The three groups (ANU, ATT and City) which derived queries from the full topic text achieved similar early precision on the VLC (0.509, 0.530 and 0.519 respectively). Groups which used only the short topic statement performed significantly worse and manually constructed queries performed significantly better.

5.2 VLC Discussion and Conclusions

The failure to more closely approach the VLC/Baseline query processing time ratio of 1.25 was almost certainly because disk storage was centralised on one of the nodes rather than distributed.

Improvement in query-processing rate may be achievable through application of some of the following additional optimisations:

1. Limit the number of document accumulators (and continue to processing terms in order of increasing *df*). This should improve memory reference locality and dramatically reduce the cost of the ranking sort.

2. Arrange the inverted file indexes in order of increasing *df* to maximise memory residency of the compressed index entries.

3. Improve the scalability ratio by using distributed disks rather than a centralised RAID box and ensuring good load balance.

4. Improve the queries. The best query processing rate among the seven runs was achieved using short, high-quality (manually generated) queries.

5. Study the relative costs of index entry decompression and disk I/O. If the former is expensive relative to the latter, query processing may be speeded up by using uncompressed or partly compressed indexes.

6. Remove term position information from the indexes. This information was not used in processing either the VLC or baseline queries. Removing it could reduce memory demands and dramatically speed up decompression of postings lists.

Improvement in the bang-per-buck measure will result from improvement in query-processing speed and/or from the use of cheaper hardware. It is interesting to note that the proposed use of local disks rather than a centralised RAID on the cluster of Alphas would reduce rather than increase the cost of the system. It is likely that the use of collection-selection techniques could dramatically improve bang-per-back performance without significantly harming early precision on the large collection.

The indexing rate of 1.29 gB/elapsed hour achieved on the VLC using a single 128 MB workstation is quite satisfactory given that the input text remains in compressed form, that the index contains full position information and that total disk space requirements (including temporary files) only amount to one third of the raw text size. With local disks on each of the eight Alpha nodes it should be possible to increase the indexing rate by close to a factor of eight, depending upon degree of load balance achievable, to over 10 gigabytes/elapsed hour.

It is planned to investigate the reason for the higher early precision observed with the larger collection in future work.

6 Conclusions

The pseudo relevance feedback method proposed here has been shown to produce consistent average benefit for all the sets of topics on which it has been tried. However, the benefit gained on the TREC-6 Adhoc tasks was not as great as that observed in training with earlier TREC tasks.

Combined with a Cornell variant of City University's Okapi BM25 scoring function, this feedback method was used very successfully in the Long-topic Automatic Adhoc category, achieving best average precision. The same run achieved best recall and best precision@20 of all 57 official Automatic Adhoc runs and was surpassed by only one run on average precision.

The same basic automatic method was used as the basis for successful submissions in the Manual Adhoc, Filtering and VLC tasks. Starting from automatically generated queries, a relatively naive PADRE97 user was able to achieve third best results in the Manual Adhoc category with only a relatively small investment of time, despite interacting with only part of

the TREC-6 data set. The effectiveness of manual refinement indicates that there is scope for improvement in the automatic query generation process. In Filtering, a Generalised Reduced Gradient non-linear optimisation method was used to set score thresholds. Only two groups achieved better official results on the F2 utility measure and three on F1.

A new BM25-based method of Concept scoring was shown to produce a small but significant gain in precision on the Manual Adhoc task. Further work is needed to prove (and hopefully improve) its usefulness. The automatic generation of concepts suitable for use in Concept and Distance scoring remains a goal of future research.

Bibliography

HAWKING, D. 1997a. PADRE for COWs. In P. MACKERRAS Ed., *Proc. Sixth Parallel Computing Workshop, PCW97* (Canberra, Australia, September 1997). Department of Computer Science, ANU. paper P1-B.

HAWKING, D. 1997b. Scalable text retrieval for large digital libraries. In C. PETERS AND C. THANOS Eds., *Proc. First European Conference on Digital Libraries*, Volume 1324 of *Lecture Notes in Computer Science* (Pisa, Italy, September 1997), pp. 127–146. Springer.

HAWKING, D. AND THISTLEWAITE, P. 1996. Relevance weighting using distance between term occurrences. Technical Report TR-CS-96-08, Department of Computer Science, Australian National University, `http://cs.anu.edu.au/techreports/1996/index.html`.

HAWKING, D., THISTLEWAITE, P., AND BAILEY, P. 1996. ANU/ACSys TREC-5 experiments. In E. M. VOORHEES AND D. K. HARMAN Eds., *Proc. Fifth Text Retrieval Conference (TREC-5)* (Gaithersburg, MD, November 1996), pp. 359–376. U.S. National Institute of Standards and Technology. NIST special publication 500-238.

ROBERTSON, S. E. 1990. On term selection for query expansion. *Journal of Documentation 46*, 4, 359–364.

ROBERTSON, S. E., WALKER, S., HANCOCK-BEAULIEU, M., AND GATFORD, M. 1994. Okapi at TREC-3. In D. K. HARMAN Ed., *Proc. Third Text Retrieval Conference (TREC-3)* (Gaithersburg, MD, November 1994). U.S. National Institute of Standards and Technology. NIST special publication 500-225.

SAVOY, J. 1997. Statistical inference in retrieval effectiveness evaluation. *Information Processing and Management 33*, 4, 495–512.

SINGHAL, A., SALTON, G., MITRA, M., AND BUCKLEY, C. 1995. Document length normalization. Technical Report TR95-1529, Department of Computer Science, Cornell University, Ithaca, NY 14853.

Experiments in Spoken Document Retrieval at CMU

M. A. Siegler, M. J. Witbrock, S. T. Slattery*
K. Seymore, R. E. Jones, and A. G. Hauptmann
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

*Justsystem Pittsburgh Research Center
4616 Henry St.
Pittsburgh, PA 15213

Abstract

We describe our submission to the TREC-6 Spoken Document Retrieval (SDR) track and the speech recognition and the information retrieval engines. We present SDR evaluation results and a brief analysis. A few developments and experiments are also described in detail including:

- Vocabulary size experiments, which assess the effect of words missing from the speech recognition vocabulary. For our 51,000-word vocabulary the effect was minimal.
- Speech recognition using a stemmed language model, where the model statistics of words containing the same root are combined. Stemmed language models did not improve speech recognition or information retrieval.
- Merging the IBM and CMU speech recognition data. Combining the results of two independent recognition systems slightly boosted information retrieval results.
- Confidence annotations that estimate of the correctness of each recognized word. Confidence annotations did not appear to improve retrieval.
- N-best lists where the top recognizer hypotheses are used for information retrieval. Using the top 50 hypotheses dramatically improved performance in the test set.
- Effects of corpus size on the SDR task. As more documents are added to the task, the gap between perfect retrieval and retrieving spoken documents gets larger. This makes it clear that the size of the current TREC SDR track corpus is too small for obtaining meaningful results.

While we have done preliminary experiments with these approaches, most of them were not part of our submission, since their impact on the IR performance on the actual TREC SDR training corpus was too marginal for reliable experiments.

1. The SDR Data and Task

The speech data for the 1997 TREC spoken document retrieval track consisted of about 70 hours of broadcast news mostly from CNN and NPR shows. The data had been segmented into stories and manually transcribed. There were three "versions" of the data available: A manually generated transcript (which also contained some errors), a speech recognition transcript provided by IBM, and the raw audio data, to be transcribed by our own recognizer. About 35 hours of this corpus was classified as training data, which we used to train the Sphinx-III speech recognition system. The remainder was held out as unseen test data. There were about 1200 stories in the training data set and 1451 in the test set. To develop and debug the system, there were 5 training queries available and the test data consisted of 49 queries.

Scoring Metrics

The test queries were designed to simulate a *known item retrieval* task. For each query, there was only one document considered relevant for the purposes of this evaluation. While other documents may have some relevance to the query, only the document it was designed to retrieve was scored as a correct retrieval. To reflect the nature of this task, we used the following metric:

Inverse Average Inverse Rank (IAIR)

$$\text{IAIR} \equiv \frac{1}{\sum_i \left(rank_i^{-1}\right)}$$

Where $rank_i$ is the rank of document i

One characteristic of the IAIR is that it rewards correct documents near the top more than documents in the middle or towards the end of the rankings. In our opinion this is a reflection of desired behavior in an IR system, and we used the metric exclusively in our analysis.

Idiosyncrasies of Known Item Retrieval

One of the idiosyncrasies of the known item retrieval paradigm is that only one document is defined to be relevant to the query. Therefore, it is in the interest of the IR system to maximize the score for this document, rather than maximize the overall number of relevant documents retrieved. As a consequence, we found that query expansion did not produce a better IAIR score. In addition, the IR system performed better when as many of the query terms as possible appeared in the correct document, despite the presence of erroneously recognized query terms in the incorrect documents. Generally, known item retrieval seems to favor the detection of correctly identified query terms over the rejection of falsely identified query terms and this is demonstrated in our experiments below.

2. System Overview

In this section we give a system description of the actual CMU TREC-6 SDR submission. The speech recognition system is outlined as well as a fully automatic information retrieval weighting scheme suitable for retrieving documents transcribed (with errors) by automatic speech recognition.

The Speech Recognition Component

The Sphinx-III speech recognition system was used for the CMU TREC SDR evaluation, and it was configured similar to the 1996 DARPA CSR evaluation [10], although several changes have been made since then. Sphinx-III is a large vocabulary, speaker independent, fully continuous hidden Markov model speech recognizer with separately trained acoustic, language and lexical models.

For the current evaluation a gender-independent HMM with 6000 senonically-tied states [5] and 16 diagonal-covariance Gaussian mixtures was trained on a union of the CSR Wall Street Journal corpus and the 1996 TREC-6 training set.

The decoder used a Katz-smoothed trigram language model trained on the 1992-1996 Broadcast News Language Modeling (BN LM) corpus. This is a fairly standard language model, much like those that have been used in the DARPA speech recognition community for the past several years. As a space optimization singleton trigrams and bigrams were excluded. As a new feature, this language model incorporated cross-sentence-boundary trigrams to better model utterances containing more than one sentence.

The lexicon was chosen from the most common words in this corpus, and to be a size that balances the trade-off between leaving words out-of-vocabulary and introducing acoustically confusable words [9]. For this evaluation, the vocabulary was comprised of the most frequent 51,000 words in the BN LM corpus, supplemented by some 200 multi-word phrases and some 150 acronyms. The vocabulary size was initially based on our experience with broadcast news, and a subsequent careful analysis of the trade-offs showed that our choice was a very good one. More details of the trade-off involved in vocabulary selection are provided below.

In contrast to the earlier Sphinx-II speech recognition system, Sphinx-III boasts a higher accuracy but at significant cost. To achieve a lower word error rate of 27.4% versus 45.9% for Sphinx-II on a subset of the training data, the original Sphinx-III system processing time increased to 120 times real time on a 266 MHz DEC Alpha compared with only 1.4 times real time for Sphinx-II. By reducing the beam width of the search and optimizing the space required, we reduced the Sphinx-III processing time to about 30 times real time, with only a slight loss in word transcription accuracy. Decoding the audio files in the test data thus required about 1000 hours of CPU time.

The Information Retrieval Component

Both documents and queries were processed using the same conditioning tools, namely noise filtering, stopword removal, and term stemming:

- Noise Filtering: The goal of noise filtering was simply to remove non-alphabet ASCII characters, punctuation, and other junk considered irrelevant to IR. All punctuation was removed except for spelled-letter words, e.g. "C. M. U," and the use of the apostrophe for contractions, e.g. "CAN'T." Any changes in case were removed.
- Stopword removal: A set of 811 stopwords was compiled from a combination of the SMART IR engine and several selected by hand based on document frequency. These words were removed entirely.
- Term mappings: A set of 4578 mappings was used to map words with irregular word endings that were not properly covered by an implementation of the Porter [7] algorithm. An on-line Houghton-Mifflin dictionary was used for this lookup of irregular words and their roots.
  An example of this mapping is `APPENDICES→APPENDIX`
- Term stemming: An implementation of the Porter algorithm was applied to map words to their common root.

A heavily stripped down core of the CMU Informedia SEIDX engine was used to compare queries with documents. A relevance score was created for each pair according to the following equation:

$$\text{Relevance Score} \equiv \frac{\sum_i (qtf_i * dtf_i * \log(idf_i))}{\sqrt{\sum_i dtf_i^2}} * \left[1 + \left(\sum_i \text{sign}(qtf_i * dtf_i)\right)^2\right]$$

qtf_i	Query term frequency for vocabulary word i
dtf_i	Document term frequency for vocabulary word i
idf_i	Inverse document frequency for vocabulary word i
sign	Sign of value function (0 if 0, 1 if positive)

3. Official TREC-6 SDR Results

Table 1 shows the official CMU TREC SDR results. Since the transcriptions were subject to filtering as discussed above, the word error rates are reported for both the unfiltered and filtered references and hypotheses. An analysis of the results showed several preprocessing errors and confirmed an insight into the relationship between word error rate and information retrieval.

Transcription Source	WER		IAIR
	Unfiltered	Filtered	
Reference	0	0	1.35
CMU-SR	35.5	26.4	1.44
IBM-SR	45.6	47.4	1.64

Table 1: Performance of the CMU TREC-6 SDR Evaluation System

Vocabulary Coverage

The words that were in the queries but were missing from the speech recognizer's 51,000 word vocabulary were "CIA", "TORCHED?", "SMOKING?", "WELL_KNOWN", and "GOLDFINGER". These problems are primarily due to inconsistencies in the preprocessing phases. While "C.I.A." was in the vocabulary, "CIA" was not, resulting in a completely missed word during information retrieval. Similarly, an oversight in the preprocessing phase allowed the question mark to become part of the word in "torched?" and "smoking?". For "well-known", each of the component words "well" and "known" were in the vocabulary, but the compound "well-known" was not there as a single token, and thus was treated as an irretrievable word. The only true missing word in our 51,000-word vocabulary was "Goldfinger". Thus the 51,000-word vocabulary selection provided excellent coverage for this test evaluation.

Recognition Accuracy versus Information Retrieval Quality

The official results confirm that vastly reduced word errors rate translates into slight improvements in information retrieval. Comparing the performance on the IBM speech recognition data to the CMU speech recognition, on the filtered texts, we find that nearly **doubling** the word error rate led to only a 14% decrease in information retrieval quality.

4. Experiments

In order to create meaningful experiments with the TREC-6 training data, 1167 documents were selected from the set and headlines were generated for 374 of them by hand. In addition, a much smaller test set composed of 103 broadcast news stories from a privately collected corpus was acquired to investigate ideas involving the speech recognition configuration. We shall refer to this latter test set as the "small test set."

4.1. Vocabulary Size Experiments

Prior to the evaluation we attempted to find a good vocabulary size that was optimized for both speech recognition and information retrieval. We chose three different vocabulary sizes, 40,000, 51,000 and 64,000 words, constructed a language model for each one, and then performed speech recognition. Table 2 shows that as the vocabulary got larger, the rate of out-of-vocabulary words decreased, but beyond 51,000 words speech recognition accuracy did not improve. Additional vocabulary coverage was thus obtained at the cost of adding many acoustically confusable words, and information retrieval effectiveness decreased slightly. We chose to use the 51,000-word vocabulary for our official submission, resulting in only one query word in the final 49 test to be missing.

Vocabulary Size	Out Of Vocabulary Rate	Word Error Rate	IAIR
40k Words	1.13 %	26.4 %	1.24
51k Words	0.83 %	26.8 %	1.21
64k Words	0.75 %	26.8 %	1.22

Table 2: Effect of Vocabulary Size on System Performance.

4.2. Stemmed Language Models

Using a small test set described above and the 51,000-word vocabulary, we also investigated the concept of language modeling tailored specifically to information retrieval. Since the words in the recognition output are filtered, a language model was built from a stemmed version of the LM training data. Each root word in the language model had multiple pronunciations to reflect the original words before filtering. Others have used this technique to improve language modeling when the vocabulary is open-ended or indeterminate [3].

For example, suppose the root forms of the words "recognize", "recognized", and "recognition" all map into the common root "recogni"+suffix, where suffix in this case is either "ze", "zed", or "tion". The stemmed language model would provide only one transition from the root "recogni" into words that can follow, in effect collapsing multiple paths between individual words into one path between root words. The lexicon would reflect the alternate original words as alternate pronunciation of the root word, i.e.

```
Recogni          R EH K AX G N AY Z
Recogni(2)       R EH K AX G N AY Z DD
Recogni(3)       R EH K AX G N IH SH AX N
```

The premise was that this stemmed language model would avoid much of the confusion due to acoustic variations in suffixes of words, but would aid in the correct recognition of the important roots of the words. Table 3 shows the results of these experiments. The word error rate of the stemmed language model was higher than for the baseline language model. The WER increased both if only stemmed words were counted, as well as when all original words were compared. Furthermore the information retrieval effectiveness (as measured by the inverse average inverse rank metric) also showed a decrease.

Language Model	Word Error Rate		IAIR
	Unfiltered	Filtered	
Baseline	26.8 %	22.6 %	1.17
Stemmed	35.1 %	23.8 %	1.25

Table 3: Using a language model built from stemmed LM training texts.

4.3. Merging Multiple Sources of Speech Recognition Data

Since the IBM speech recognition system was developed independently of the CMU system, and it used different training data, vocabulary, and language models, it occurred to us that a combination of the two speech recognition transcripts might allow some randomly distributed errors to be recovered. Instead of mixing the recognition outputs, we formed a weighted relevance score in the following way:

$$Score_{MIX} \equiv Score_{CMU} * \lambda + Score_{IBM} * (1 - \lambda)$$

$Score_{CMU}$	The relevance score using the CMU recognition output
$Score_{IBM}$	The relevance score using the IBM recognition output
λ	An interpolation weight

Results on the TREC-6 testing set are shown in Table 4, showing a slight reduction of retrieval error when the CMU weight is 0.8 and the IBM weight is 0.2. Thus multiple recognizers, even with widely varying word error rates, can be combined to improve information retrieval performance.

CMU Weight	IBM Weight	IAIR
1.0	0.0	1.382
0.9	0.1	1.379
0.8	**0.2**	**1.375**
0.7	0.3	1.395
0.6	0.4	1.394
0.5	0.5	1.421
0.4	0.6	1.467
0.3	0.7	1.462
0.2	0.8	1.467
0.1	0.9	1.548
0.0	1.0	1.581

Table 4: Results of merging relevance from separate recognition systems.

4.4. Confidence Annotation

Since state-of-the-art speech recognition software does not produce a perfect transcript of what was said, we would like to obtain any extra information we can about the likelihood of correctness of particular words. This is akin to a human annotator guessing a mumbled word and indicating a possible transcription error.

An ideal automatic confidence annotator would label each word produced by the speech recognizer with a label *correct* to indicate that this is in fact the word that was spoken, and *incorrect* to indicate that this word was not spoken. We will compare the results of our annotation to this ideal, which we call Perfect Annotation.

Features for Confidence Annotation

The confidence annotation we performed is based on work by Lin Chase [1], though annotation has been explored by many others including [2][3][4]. Typically confidence annotation is performed by taking information available about individual occurrences of words in the hypothesized text, from information produced within the speech recognizer, or outside the recognizer. These features are then automatically examined to find indicators of likely correctness and incorrectness.

The candidate features we considered were:

- *Acoustic Score.* This is the score the speech recognizer assigns the word based the probability that the acoustics observed were generated by the hypothesis.
- *Language Model Score.* This is a score assigned by the speech recognizer, based the probability that the word is to occur given the previous two words.
- *Duration.* This is the duration of the word, and helps offset the duration dependence of the acoustic score.

- *N-best Homogeneity.* The N-best list is the list of the best n guesses at the words spoken in the document, sorted according to a weighted combination of acoustic and language model scores. A word appearing in our hypothesis may appear in many or few of the competing hypotheses. N-best list homogeneity is the proportion of hypotheses that the word appears in. We set n to 200 for the confidence annotation experiments.

Experimental Description - Confidence Annotation

For each set of features, the experiment proceeds as follows:

- Label all words in training set as *correct* or *incorrect*[1] by comparing them to the words in the words in the reference transcript
- Build a decision tree that finds sets of features that perform well in distinguishing between *correct* and *incorrect* words in speech recognition hypotheses.
- Use decision tree to test features of words in test set. Once a word has been sorted into a leaf node, the proportion of correct and incorrect words from the training set with these features is used to calculate an approximate probability of correctness
- Perform information retrieval by weighting each word according to the probability that it is correct (the *confidence*).

We conducted experiments by splitting the training data into two sections, training our decision tree on one half, testing on the other half, then reversing the roles.

Decision Tree Building

The decision tree building algorithm we use is C4.5 [8]. It functions by taking all training data, and attempting to find rules based on features which distinguish between classes. Each item of training data is a word along with its associated features (described above), and its class of *correct* or *incorrect*. Taking each feature does this in turn, asking a question about that feature, and using the answer to partition the data. A feature is chosen if it has high information gain, i.e. if the resulting two groups of data contain less of a mix of *correct* and *incorrect*. The ideal split would create classes that contain exclusively *correct* or exclusively *incorrect* examples.

Since such ideal splits are rare, the decision tree building halts when no more information gain (reduction in entropy) can be achieved. At this point, each leaf of the tree contains examples which have all the same features for questions asked at each partition, and which are mostly of one class. The proportion of *correct* examples at this node is the probability of correctness that will be assigned to any word with the same features.

When using the decision tree to classify a new word, we check each of its features to find which leaf-node of the decision tree to classify it into. At that point, it is classified as having the probability of correctness corresponding to this leaf node.

Evaluating Confidence Annotation: Cross-Entropy Reduction

The most common method of evaluating word confidence annotation is cross-entropy reduction. Cross-entropy is a measure of how well our model of the probability of word correctness corresponds to Perfect Annotation (as defined above). If our model annotates perfectly, its cross-entropy is 0. The worse the annotation performs the higher the cross-entropy.

---

[1] incorrect words are all insertions and substitutions in the hypothesis

The most naive form of confidence annotation we can perform is to tag each word with a probability of correctness equal to the overall word-accuracy. Thus if we know that our recognizer generally gets 80% of words correct, the baseline confidence annotator assigns each word an 80% probability of correctness. We then measure the quality of our annotation by measuring how much better it performs than this baseline.

$$CrossEntropy \equiv \frac{1}{n}\sum_{i}^{n} P(w_i) * \log_2 \frac{1}{Q(w_i)}$$

$P(w_i)$ The actual probability that word *i* is incorrect

$Q(w_i)$ The probability that word *i* is incorrect as predicted by the annotation

Thus we attain a figure for cross-entropy for the default model of classifying each word as correct with probability equal to the word-accuracy, and score our improvements in modeling the probability of correctness by how much they reduce cross-entropy as a percentage of this baseline.

Information Retrieval Using Word Confidence Weights

First we describe two orthogonal ways of using word confidence weights in the relevance scheme described above:

- **Expected Term Frequency** (ETF): The ETF is an estimate of how many times the term actually occurred given the number of observations. Assuming independent observations, this is a sum of the probability of a word being correct over each instance.

- **Expected Inverse Document Frequency** (EIDF): To calculate EIDF, we first calculate the probability that this word occurs somewhere in the document, for each document:

$$\begin{aligned} P(w \in d) &= 1 - P(w \notin d) \\ &= 1 - \prod_{i}^{n} P(w \neq w_i) \\ &= 1 - \prod_{i}^{n} [1 - P(w = w_i)] \end{aligned}$$

Since typically, $P(w = w_i)$ is very small when $w \neq w_i$, we only take the product over terms for which the recognized word was *w*. Summing this value over all documents and dividing by the total number of documents gives us an approximate value of the expected document frequency for this word

Oracle Experiments

Since the interaction between confidence annotation and information retrieval may be complex, we also conducted an experiment to see how we could make use of confidence scores in the idealized case in which we know exactly which words are correct, and which are incorrect. We removed words in two different ways:

- Pre-filter: Before the hypothesis is filtered, all the words that are not found in the reference are removed.
- Post-filter: After the hypothesis is filtered, all the words that are not found in a filtered version of the reference are removed

Table 5 shows that for both training and testing sets, the Post-Filter Oracle annotation was able to significantly reduce the IR error of the decoded transcripts. This indicates that a more realistic experiment might be able to do this as well.

We performed an analysis of some of the differences between documents in the stemmed oracle experiment, and reference information retrieval experiments. We should expect the number of query words in the correct document to decrease, since oracle confidence annotation cannot *correct* for substitutions and deletions, but will drop all incorrectly substituted and inserted words. A cursory glance at documents and queries revealed that some documents contain **more** query words as speech hypotheses then the corresponding reference transcription. Our intuition here is that speech recognition can occasionally correct for spelling errors in the references, and so words that are incorrect with respect to the reference transcription may be correct for the purposes of information retrieval.

	Baseline Performance		Oracle Annotation	
	Reference Transcripts	Speech Transcripts	Pre-Filter	Post-Filter
Training Set	1.233	1.283	1.285	1.269
Testing Set	1.332	1.382	1.374	1.338

Table 5: Baseline and Oracle Annotation on TREC-6 Training and Testing Sets. Values are IAIR

Information Retrieval Experiments for Confidence Annotations

In order to see how well cross-entropy reduction translates into gains in information retrieval accuracy, we conducted a series of experiments. Since we also hoped to find the best way of incorporating weights into information retrieval we performed the following information retrieval experiments:

- **ETF**: for this experiment, we used ETF, and regular IDF.
- **EIDF**: for this experiment, we used EIDF, and regular TF.
- **ETFIDF**: we use both ETF and EIDF

	Pre-Filter			Post-Filter		
	ETF	EIDF	ETFIDF	ETF	EIDF	ETFIDF
Training set	1.276	1.283	1.277	1.273	1.281	1.274
testing set	1.378	1.383	1.399	1.381	1.382	1.382

Table 6: Confidence Annotation Performance on TREC-6 Training and Testing Sets. Values are IAIR.

The results of these experiments are found in Table 6. Although the IAIR was reduced in most cases, the upper bound found in the Oracle Annotation was not attained.

4.5. Using N-best Lists for Information Retrieval

Typically, speech recognition systems produce a transcription of each spoken utterance in much the same way that a human transcriber might. However, the transcription used is only the most probable decoding of the acoustic signal, out of a large number of hypotheses that are considered during the recognition process. It is a relatively simple matter to obtain a list of these different hypotheses, ranked in order of decreasing likelihood.

Using these additional hypotheses seems promising for information retrieval, since it offers the hope of including terms that would otherwise be missed by the speech recognizer in documents, allowing them to match with query terms and increase document recall. On the other hand, words incorrectly identified in lower ranked recognition hypotheses may cause spurious matches with query terms, decreasing retrieval precision.

Experiments Using N-Best Lists

In the context of the TREC-6 SDR task, an initial attempt was made to evaluate retrieval effectiveness using N-best hypotheses lists generated from the speech recognition decoder lattice. N-Best hypotheses were generated for the 1451 stories in the TREC-6 SDR test data. Of these, decoding failed completely in four cases, resulting in empty transcriptions. For the remaining 1447 stories, lists of the two hundred most likely hypotheses were generated for each utterance. Table 7 shows an exampie of N-best hypotheses.

Ideally, one would use hypothesis probabilities generated during decoding to weight the terms during retrieval, but for this preliminary experiment, the N hypotheses for each utterance were simply concatenated together into one larger document. No discounting of weights for less probable hypotheses was done.

N	Nth most likely decoder hypothesis
1	HATE FAIR ADEQ EDUC CHILD WITHSTAND CALM
2	HATE FAIR ADEQ EDUC CHILD WITHSTAND COMMON
3	HATE FAIR ADEQ EDUC CHILD WITHSTAND INTERCOM
4	HATE FAIR ADEQ EDUC CHILD WITHSTAND CALM

Table 7: The top four hypotheses for utterance three of story j960531d.7, after stop word removal and stemming. Note that the fourth hypothesis is identical to the first, and differed only in inflected forms.

The effect on retrieval effectiveness of using the documents generated from the N-best lists in the TREC-6 test set is illustrated in Table 8. Note that for N set to 50, the performance on the hypothesized transcripts is actually slightly lower than performance on the reference transcripts (1.332) This may be again due to effects of misspellings in the reference transcripts.

Number of Hypotheses (n)	1	2	5	10	20	50	100	200
IAIR	1.368	1.353	1.366	1.365	1.367	1.317	1.320	1.325

Table 8: IR Performance of N-Best hypotheses on the TREC-6 test set.

While it is encouraging that an improvement in retrieval can be obtained at all by this method, it is clear that further work will be required if the promise of this idea is to be realized. In particular, the increasingly harmful effect of adding large numbers of less probable hypotheses to the documents suggests that discounting each hypothesized word by its recognition score may improve performance even more.

4.6. Scaling The Collection Size

Many of our experiments, including some of the ones reported here, seem to suffer from two problems. The effect size of our experimental variables seems to be fairly small, and the difference between the reference text retrieval and the speech recognition transcript retrieval is only a few percent of the inverse average inverse rank. If this relationship holds even as we scale to larger, more realistic, and more useful collections, then we can consider the problem of spoken document retrieval practically solved to within a few percent of perfect text retrieval effectiveness.

To test this hypothesis using the TREC-6 training set, we increased the number of text documents in the corpus up to 14,000 and measured the inverse average inverse rank for the same retrieval queries. However, instead of actually performing speech recognition on the added documents, artificially degraded texts were used. In this case, the degradation method attempted to only model word errors through deletion of query words. Although a primitive model of speech recognition errors this may represent an upper performance bound.

Figure 1 shows the relationship between the inverse average inverse rank information retrieval performance and the size of the document collection. As more documents are added to the collection, the gap between the reference (perfect text) retrieval and the speech recognition based retrieval grows. At collections larger than 10,000 documents the gap starts to widen significantly. We can expect to experience larger discrepancies between speech transcribed and perfectly transcribed documents, which may make spoken document recognition unusable for collections numbering in the 100,000 or larger.

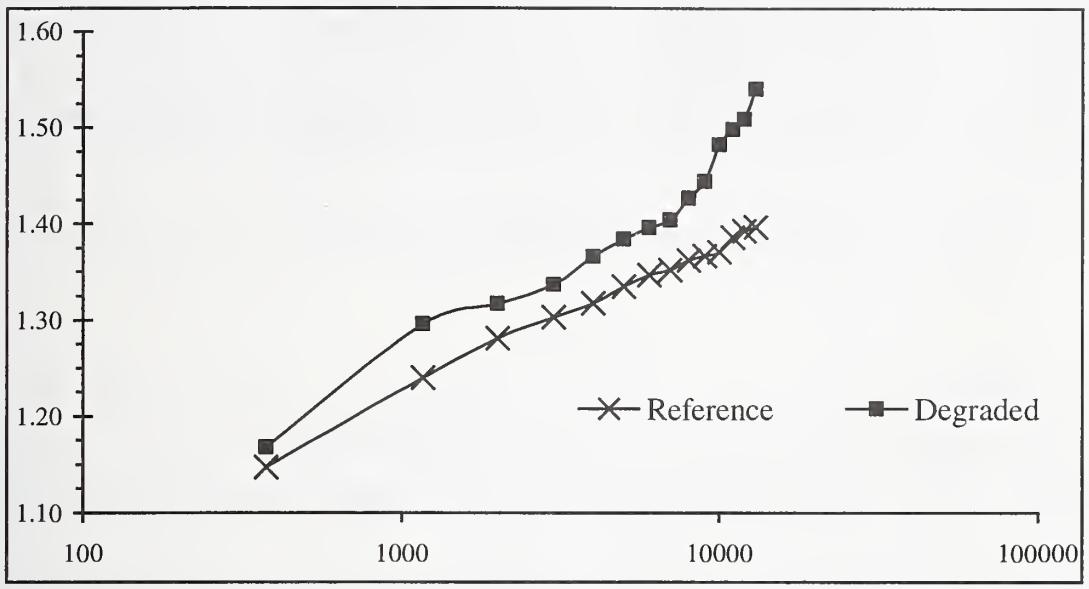


Figure 1: Effect of collection size on IR performance of the TREC-6 training set with reference and artificially degraded documents. The X Axis is the number of documents used in the analysis, and the Y Axis is the IAIR.

5. Summary

There are several conclusions we can draw based on our experiments:

- First of all, we have found that even large reductions in speech recognition word error rate result only in small information retrieval improvements. On the converse side, the quality of information retrieval is a lot higher than the speech recognition word error rate figures would indicate. Despite fairly high word error rates, information retrieval performance was only slightly degraded for speech recognizer transcribed documents.

- Stemmed language modeling did not help speech recognition or information retrieval.

- A 51,000 vocabulary covered the range of words used in the queries quite well. Only one query word was truly outside of this vocabulary.

- We could expect better performance on the reference texts if better IR weighting schemes and pre-processing functions were used. These improvements would probably also result in small gains in the speech corpus, although we have done no studies.

- Confidence Measures provide no benefit. Even an oracle confidence measure, which can reliably single out the correctly recognized words and discard all the other words provides only a small increase in retrieval effectiveness (as measured in IAIR). This points to the conclusion that deleted (missing) words are most critical, while inserted words do not affect the retrieval in the same proportion.

- Since deleted (missing) words are critical to the retrieval effectiveness, one can try to reduce this by adding probable words from the speech recognizer hypothesis N-best list. Using the N-best list to augment the speech recognition output with likely words shows great promise. Our experiments indicate that this approach might drastically reduce the difference between perfect text transcripts and speech recognizer generated transcripts.

- Merging the results from multiple independent speech recognizers may also improve IR effectiveness.

In general, most of our findings are very preliminary. While we believe we may have uncovered trends, there is too little data for conclusive experiments. As a result, we did not conduct significance tests to measure the practical effects of the observed trends since the TREC-6 SDR track provided too little data for definitive experiments. Furthermore, the difference between the speech recognizer generated transcripts and the perfect text transcripts was too small in this corpus. However, the experiments we have done on increasing the scale of these document collections by orders of magnitude leave a worrisome fear that the initially promising results for SDR will not hold up in larger data sets.

We have viewed this participation in the TREC-6 SDR track as a learning experience, which will guide both our own research as well as the design of future SDR track evaluations.

References

[1] L. Chase, *PhD thesis*, Carnegie Mellon University Robotics Tech Report, 1997.

[2] S. Cox and R. Rose, "Confidence Measures for the Switchboard Database," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1996.

[3] P. Geutner, "Using Morphology Towards Better Large-Vocabulary Speech Recognition Systems," *IEEE International Conference on Acoustics, Speech and Signal Processing, 1995.*

[4] L. Gillick and Y. Ito, "Confidence Estimation and Evaluation," *LVCSR Hub-5 Workshop Presentation*, 1996.

[5] M-Y. Hwang, "Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition". PhD Thesis, CMU-CS-93-230, Carnegie Mellon University, 1993.

[6] P. Jeanrenaud, M. Siu, H. Gish, "Large Vocabulary Word Scoring as a Basis for Transcription Generation," *Proceedings of Eurospeech,* 1995.

[7] M. F. Porter, "An algorithm for suffix stripping," *Program*, 14(3):130-137, July 1980.

[8] J. R. Quinlan, *Programs for Machine Learning*, San Francisco, Calif.: Morgan Kaufmann, 1993.

[9] K. Seymore, S. Chen, M. Eskenazi, and R. Rosenfeld. "Language and Pronunciation Modeling in the CMU 1996 Hub 4 Evaluation," *Proc. Spoken Language Systems Technology Workshop*. Morgan Kaufmann Publishers, 1997.

[10] M. Siegler, U. Jain, B. Raj, and R. Stern. "Automatic Segmentation, Classification, and Clustering of Broadcast News Audio," *Proc. Spoken Language Systems Technology Workshop*. Morgan Kaufmann Publishers, 1997.

Acknowledgments

This research was supported in part by DARPA under research contract F33615-93-1-1330 and N00039-91-C-0158. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or the U. S. Government. We would like to thank Ravi Mosur, Eric Thayer, and Stan Chen for their invaluable contributions to this work.

Passage-Based Refinement (MultiText Experiments for TREC-6)

Gordon V. Cormack[1] Charles L. A. Clarke[2] Christopher R. Palmer[1] Samuel S. L. To[1]

MultiText Project
[1] Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada
[2] Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada
mt@plg.uwaterloo.ca
http://multitext.uwaterloo.ca

Abstract

The MultiText system retrieves passages, rather than entire documents, that are likely to be relevant to a particular topic. For all runs, we used the reciprocal of the length of each passage as an estimate of its likely relevance and ranked accordingly. For the manual adhoc task we explored the limits of user interaction by judging some 13,000 documents based on retrieved passages. For the automatic adhoc task we used retrieved passages as a feedback source for new query terms. For the routing task we estimated probability of relevance from passage length and used this estimate to construct a compound (tiered) query which was used to rank the new data using passage length. For the Chinese track we indexed individual characters rather than segmented words or bigrams and used manually constructed queries and passage-length ranking. For the high precision track we performed judgements on passages using an interface similar to that used for the manual adhoc task. The Very Large Collection run was done on a network of four cheap computers using very simple manually constructed queries and passage-length ranking.

1 Introduction

The MultiText Project participated in the routing and adhoc tasks, and in the Chinese, high precision and very large collection tracks. For the adhoc task we submitted both automatic and manual runs; for the routing task and the tracks we submitted manual runs. These experiments explored a variety of methods for query expansion and refinement based on arbitrary passage retrieval.

The major research focus of the the MultiText Project is the development and prototyping of scalable technologies for distributed information retrieval systems. The MultiText system is based on the federated architecture shown in Figure 1. The system is composed of several major components: The *index engines* maintain the index file structures and provide search capabilities. The *text servers* are specialized by document type and provide retrieval capabilities for arbitrary text passages specified at the word level. Finally, the *marshaller/dispatcher* interacts with clients and coordinates query and update activities.

Research issues are addressed in the context of this distributed architecture. Issues of concern to the MultiText Project include data distribution, load balancing, fast update, compression, fault tolerance, document structure, ranking, and user interaction. Support for document structure is a

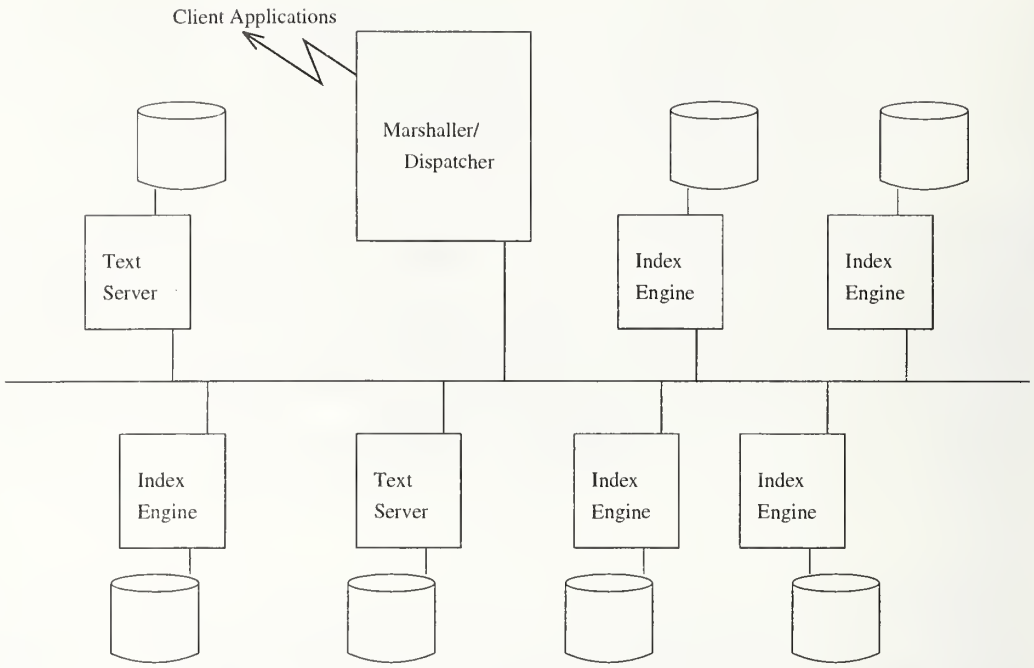


Figure 1: Architecture of the MultiText System

particular feature of the MultiText System. The system can support multiple document formats within a single integrated database and provide specific support for structure inherent in each document type. The MultiText query language, GCL, provides facilities for directly referencing document structure and allows queries to reference equivalent structural elements across differently formatted documents [2]. Ranking in the MultiText system is based on passage retrieval, with the score of a passage based on its length, and the score of a document based on the score of the passages contained within it. As well as full documents, the method allows ranking of arbitrary document components. Scores do not depend on collection-wide statistics, making the ranking method particularly suitable for use in a distributed environment.

Our TREC-4 paper introduced the basic ranking technique, *shortest substring ranking* [3]. Our TREC-5 paper extended this work and introduced a method of passage-based interactive query expansion and refinement [1]. For TREC-6 we have extended this interactive query refinement method for participation in a wide range of tracks and tasks. In addition, for the first time, we submitted a fully automatic run that extends and validates methods introduced since TREC-5 [4].

The next section summarizes the GCL query language and shortest substring ranking. The subsequent section provides an overview of our TREC-6 work. Section 4 describes our work for each of the tasks and tracks in detail.

2 Shortest Substring Ranking

We model the text in a database as a sequence of terms. Document structure is indicated by structural markers indexed at positions between the terms. A solution to a query is a set of intervals from this text sequence. Each interval is represented by an *extent*, an ordered pair (p, q), with $p \leq q$, representing a start and end position in the text sequence.

The simplest GCL query is a phrase. The solution to the query

```
"Hubble Telescope"
```

is the set of extents corresponding to the locations of this phrase within the text sequence.

GCL supports the standard boolean operators. Each interval in the solution set for the query

```
"Africa*" and "civilian" and ("death*" or "casualties")
```

satisfies the conditions implied by the boolean operations. In order to limit its size, we restrict the solution set to include only those intervals that do not contain smaller intervals that satisfy the query conditions. This *shortest substring rule* provides linearity and ordering properties that make efficient query evaluation possible and is central to the document scoring technique. Ranking is based on the length and density of these solution intervals.

An ordering operator ("...") is provided to link the start and end positions of text intervals defined by sub-queries. The query

```
"<title>"..."</title>"
```

links the start and end tags for titles, and has as its result the set of all titles. Similarly, the query

```
"<doc>"..."</doc>"
```

has as its result the set of all documents. The shortest substring rule guarantees that the solution set contains only single documents. Start and end tags that are more than a single document apart are not linked. The language includes four *containment operators*— `containing`, `not containing`, `contained in`, and `not contained in`— which may be used to query structural relationships. The query

```
("fiber"..."globe") contained in ("<doc>"..."</doc>")
```

finds occurrences of "fiber" followed by "globe" within the same document. Note that elements in the solution set for this query are text intervals, not documents. In our queries for TREC-6 we made frequent use of the "`contained in`" operator for passage length restriction. A major element in our work for the routing task was a method for determining appropriate values for passage length restrictions. The query

```
("fiber"..."globe") contained in 6 words
```

finds occurrences of "fiber" followed by "globe" that are six words or less in length.

For ranking purposes, an extent (p, q) is assigned a score using the formula

$$I(p,q) = \begin{cases} \frac{\mathcal{K}}{|q-p+1|} & \text{if } |q-p+1| \geq \mathcal{K} \\ 1 & \text{otherwise.} \end{cases}$$

Intervals whose length is less than the "cutoff" parameter $\mathcal{K}$ are assigned a score of 1; larger intervals are scored in proportion to the inverse of their length. Our TREC-6 experiments used values of $\mathcal{K}$ between 1 and 16.

Document scores are computed by combining the scores for the solution intervals contained within them. Given a document D and a query Q, and assuming the document contains solutions

(p_1, q_1), (p_2, q_2), ..., (p_n, q_n) ordered such that $I(p_i, q_i) \geq I(p_j, q_j)$ if $i < j$, the document's score is computed using the formula

$$S(D, Q) = \sum_{i=1}^{n} \gamma^{i-1} I(p_i, q_i)$$

Where $0 < \gamma \leq 1$ is a geometric "decay" parameter. Values for γ of 1 and 0.5 were used in our TREC-6 experiments.

This scoring method may be generalized to apply to any set of text intervals, not just documents. Using this method, the solution set for an arbitrary GCL query may be ranked in terms of a second GCL query without any need for special indexing. Since there are no collection statistics and every document is independently ranked the method is particularly suited to the distributed architecture of the MultiText System.

A GCL query produces a fixed solution set, which is then ranked using the formula given above. If a larger set of ranked documents is required —as is the case of many of the TREC tasks and tracks where up to a thousand ranked documents may be submitted — secondary queries may be used to provide these extra documents. A GCL query may be augmented by an ordered set of additional query "tiers", which are used, in order, when the initial query is exhausted. For example, a tiered query for topic 10004 ("Iranian Arms to Bosnia") might be:

```
("iran*" or "tehran") and "bosnia*" and ("arms" or "weapons")
("bosnia*") and ("arms" or "weapons")
("iran*" or "tehran") and ("arms" or "weapons")
```

The query in the first tier is intended to be a precise expression of the requirements underlying the topic. Queries in later tiers are "weaker" and are intended to pick up a large number of possibly relevant documents. The use of user-constructed tiered queries proved to be of limited value in our previous TREC work [1]. However, we have developed methods for constructing tiers based on relevance information. As a consequence, we have retained the method for further exploration, particularly as an element of our work for the routing task.

3 Passage-Based Refinement

Our TREC-6 efforts represent a diversity of exploratory and comparative work. These efforts are unified by a common theme of passage-based refinement. The MultiText System is distinguished by its support for arbitrary passage retrieval: the retrieval of passages defined at query time rather than at build time.

Our TREC-5 work introduced an interactive method for query expansion and refinement based on the selection of terms from relevant passages. This method avoids any requirement for users to examine full documents by focusing attention on the portions of the documents that are likely to be of greatest interest.

This interaction method was used in most of our TREC-6 work. Our Chinese track results represent a direct application of the TREC-5 methodology to a Chinese language environment. For the manual adhoc task and the high precision track, the method was used to make fast relevance judgements.

For the automatic adhoc task we used automatic feedback to select terms from high-scoring passages as a method of query expansion.

For the routing task we developed a method of query re-structuring that depends on passage length restrictions.

4 MultiText Experiments for TREC-6

4.1 Routing

We submitted two routing runs. The baseline run (`uwmt6r0`) used queries created using the manual technique from TREC-5: a tiered boolean query was created from combinations of terms that were found to select relevant documents. An initial query was created manually from the topic statement and was refined interactively by selecting words and phrases from relevant passages. For the comparison run (`uwmt6r1`) we decomposed these queries into elementary combinations and re-tiered them to place the best combination of terms first. The queries were further refined using length restriction and further expanded using additional interaction. The methods used to create the comparison run from the baseline run were partially automatic and partially manual. Our intention is to make the process fully automatic in the future.

Our method for re-tiering uses relevance information to place a query more likely to select a relevant document in an earlier tier. For a given query Q and document D, the probability $p(D)$ that the document is relevant is assumed to be a monotone function of the score of D with respect to Q; that is, $p(D) = F_Q(S(D,Q))$. If F_Q were known, it would be possible to construct an optimal tiered query as follows:

available-queries $\leftarrow \{(Q_i, \text{BEST-SCORE})\}$
repeat until $|$*available-queries*$| = 1$
  find (Q_1, S_1) in *available-queries* such that $F_{Q_1}(S1)$ is maximized
  remove (Q_1, S_1) from *available-queries*
  find (Q_2, S_2) in *available-queries* such that $F_{Q_2}(S_2)$ is maximized
  find S_1' such that $F_{Q_1}(S_1') = F_{Q_2}(S_2)$
  add (Q_1, S_1') to *available-queries*
  next tier of query is Q_1, restricted to scores better than S_1'
end loop
only one element, (Q_i, S_i) remains in *available queries*
final tier is Q_i

We used the training data to estimate F_Q for elementary queries consisting of a simple conjunction of terms. To estimate F_Q we ranked the documents in the training set by Q and plotted the number of relevant documents retrieved as a function of the total number of documents retrieved; we use the slope of this curve as an estimate of the probability of relevance at any point. We labeled each curve with the reciprocal of exponentially decreasing score values; from the slope and these labels we were able to estimate F_Q.

Consider the example in figure 2. The initial slope of the curve labeled "q5" is approximately 0.85; at 100 documents retrieved the slope diminishes to approximately 0.5. On the other hand, the curve labeled "q3" has an initial slope of .54; at 175 documents this slope diminishes to 0.42. The slope of "q1" diminishes much more abruptly: up to 200 documents it is about the same as "q3" (0.54) but beyond this point it rapidly approaches zero. Both "q0" and "q2" have initial slopes of 0.3, which diminishes after 300 documents. Applying the re-tiering algorithm manually to this query set, we conclude that the best tiered query would be:

q5a (q5 up to the score of the 100th document)
q3a (up to the score of the 175th document)
q1a (up to the score of the 200th document)
q5b (beyond the score of the 100th document)

q3b (beyond the score of the 175th document)
q2a (up to the score of the 300th document)
q0a (q0 up to the score of the 300th document)
q4
q2b (q2 beyond the score of the 300th document)
q1b (q1 beyond the score of the 300th document)
q0b (q0 beyond the score of the 300th document)

Because of the uncertainty in the estimates, and in order to reduce the number of tiers, we combine tiers with nearly equal slopes:

q5a
q3a+q1a+q5b
q3b
q2a+q0a
q4
q2b+q1b+q0b

At present, GCL does not directly permit the score restriction described above. However, with a decay parameter of $\gamma = 0.5$, the score of document is bounded to twice the score of the best passage in the document. Furthermore, the score of a passage is based directly on its length. Therefore, the score restriction may be approximated using a length restriction. This approach was taken for our TREC-6 routing queries.

Figure 3 shows that this refinement process provided an improvement over our baseline. We found the the insight gained from our analysis of the routing task to be useful in the manual ad hoc task, as well as in the high precision and very large collection tracks. We aim to automate the refinement process in future.

4.2 Automatic Adhoc

For the automatic adhoc task our primary interest is in the title-only run, where the title of each topic is treated as the query. These queries are very short: the longest consists of four terms; the average length is less than three terms; three consist of a single term. Previous MultiText research [4] introduced a new method for ranking very short queries. The method, called *cover density ranking*, is related to the ranking method described in Section 2. Cover density ranking forms the basis for our TREC-6 automatic adhoc experiments.

Similar to the tiered ranking of Section 2, cover density ranking is a two step process. Documents are ranked first by *coordination level*, the number of distinct query terms appearing in the document. For example, given a query consisting of three terms, a document containing at least one occurrence of each term is ranked ahead of a document containing only two of the terms, which in turn is ranked ahead of document containing only one of the terms. A secondary ranking procedure is then applied to the documents within each coordination level to produce the final document order. For this secondary ranking, the score of a document is determined by the applying shortest substring ranking formula of Section 2 to the logical conjunction of the query terms appearing in the document.

Cover density ranking may be expressed in terms of the procedure of Section 2. The GCL query language supports a "generalized combination" operator with syntax

$$N \texttt{ of } (Q_1, Q_2, \ldots, Q_M)$$

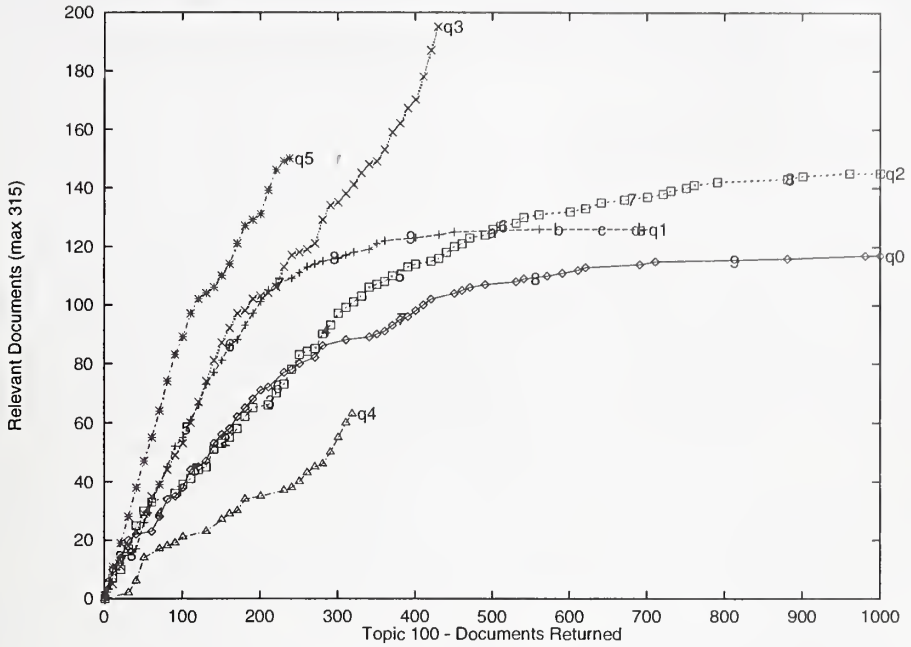


Figure 2: Slope Estimation

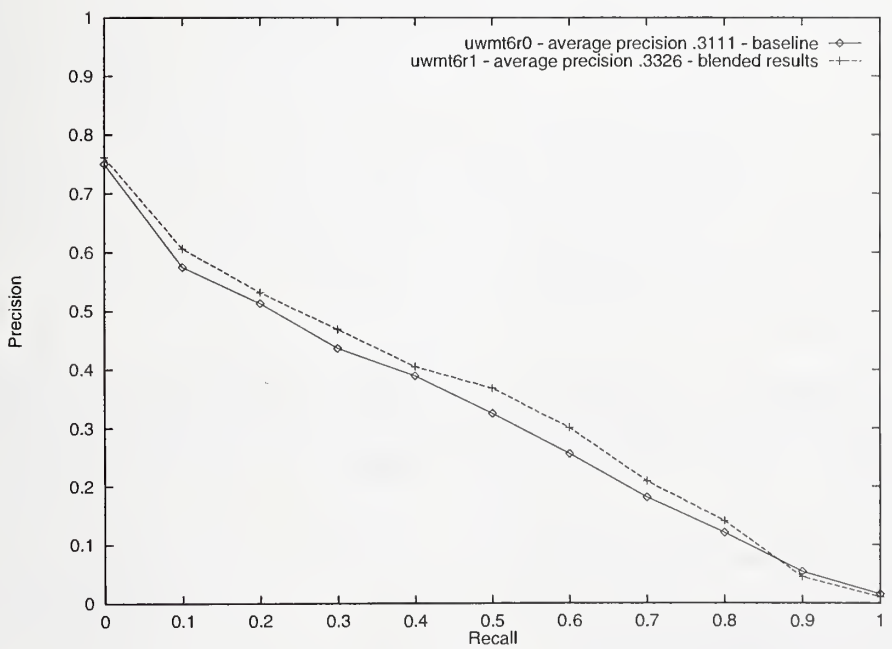


Figure 3: Routing Runs

where $Q_1 ... Q_M$ are GCL subqueries and $N \leq M$. The operator finds the set of text intervals that contain a solution to exactly N of the M subqueries under the shortest substring rule. Given query terms $T_1 ... T_M$, cover density ranking is implemented using the tiered query set

$$\begin{aligned} M &\texttt{ of } (T_1, T_2, ..., T_M) \\ M-1 &\texttt{ of } (T_1, T_2, ..., T_M) \\ M-2 &\texttt{ of } (T_1, T_2, ..., T_M) \\ &\cdots \\ 1 &\texttt{ of } (T_1, T_2, ..., T_M) \end{aligned}$$

For very short queries, a user preference for coordination level ranking has been observed by several groups [5, 8, 9]. For these queries, cover density ranking can provide retrieval effectiveness comparable to that of more traditional ranking methods [4]. Cover density ranking directly satisfies the user preference while maintaining good retrieval effectiveness. In addition, cover density ranking is fast and requires only a simple word-position index for implementation.

Our TREC-6 experiments for the automatic adhoc task had two goals: The first goal was to gain additional experience with the cover density ranking method in a comparative setting. In addition to the comparisons made possible by TREC experimental environment, we intended to directly compare cover density ranking with an established ranking method. The second goal was to provide a preliminary evaluation of local feedback based on the passages identified during the cover density ranking procedure. These passages are often short and were expected to be a valuable source of terms for query expansion.

For feedback purposes, passages were extracted using the generalized combination operator

$$N \texttt{ of } (T_1, T_2, ..., T_M)$$

using the largest value of N for which solutions existed. An upper bound of 64 words was set as a maximum passage length for feedback; passages longer than this 64 word bound were eliminated from consideration. Passages shorter than 64 words were expanded symmetrically to 64 words, with the original length retained for scoring. Words appearing in these passages were scored on the basis of their frequency of occurrence in the passages, their frequency of occurrence in the database, and the (original) length of the passages. The 24 highest scoring words were added to the query. For example, the original (title-only) query for topic 301

```
crime international organized
```

expanded into the query

```
activity combating cooperation crime crimes criminal drug enforcement fight
intelligence internal international main ministry narcotics nations
organization organizations organized organs republic sec smuggling struggle
terrorism
```

One of the goals of our TREC-6 automatic adhoc experiments was a direct comparison of cover density ranking with an established ranking method. In order to permit this direct comparison we implemented a version of the Okapi measure [6, 7] as part of the MultiText System. For a document D and a query Q we compute

$$\sum_{t \in D \wedge t \in Q} \log \left(\frac{N - n_t + 0.5}{n_t + 0.5} \right) \left(\frac{f_{D,t}}{f_{D,t} + l_D / l_{avg}} \right)$$

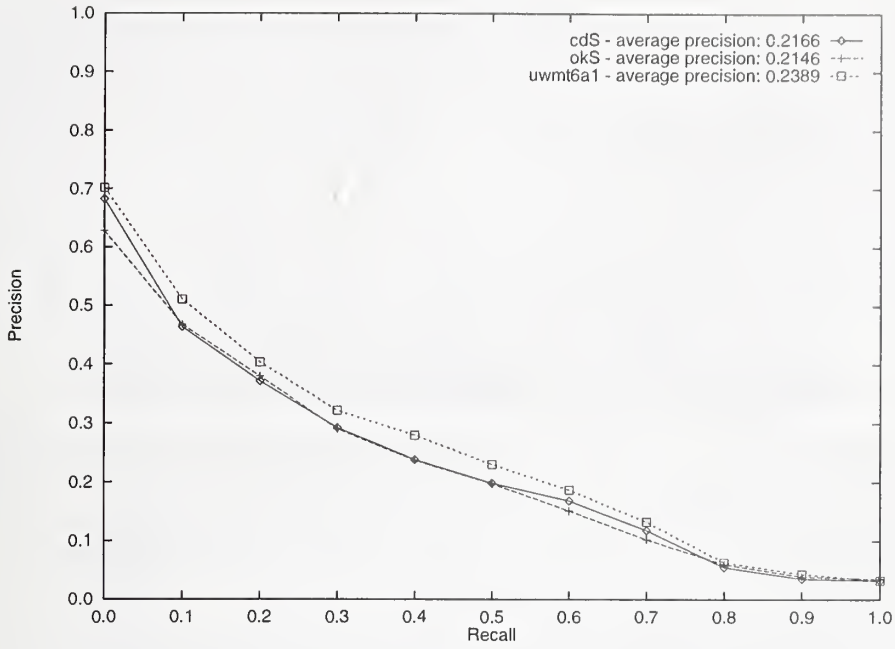


Figure 4: Automatic Adhoc Title-Only Runs

where:

$$
\begin{aligned}
N &= \text{number of documents in the collection;} \\
n_t &= \text{number of documents containing term } t; \\
f_{D,t} &= \text{frequency of ocurrence of term } t \text{ in document } D; \\
l_D &= \text{length of document } D; \\
l_{avg} &= \text{average document length.}
\end{aligned}
$$

In experiments reported by the RMIT group at TREC-4, a similar version of this measure proved to be particularly appropriate for short queries [9]. In addition, since cover density ranking is not suitable for use with the expanded queries, the implementation of this measure permitted us to combine the results of feedback with the results of the original cover density ranking.

Figure 4 presents the results of our experiments with the title-only queries. The figure plots recall-precision curves and provides average precision values for three runs. The run labeled `cdS` used cover density ranking. The run labeled `okS` used the MultiText implementation of the Okapi measure. For both of these runs the terms were stemmed, but no other query modification or expansion methods were used. The run labeled `uwmt6a1` incorporates query expansion through the local feedback procedure described earlier. For expediency, we created this run by combining the results of three runs: `cdS`, `okS` and `fbfS` (not shown). The `fbfS` run was generated by applying the Okapi measure to the fully expanded queries. For this run, the terms were not stemmed. To create `uwmt6a1`, the document scores in `fbfS` and `okS` were first normalized and added, and the result was then combined with `cdS` by adding ranks. We submitted `uwmt6a1` as an official TREC-6 run; the run was not judged.

As expected, the performance of cover density ranking was comparable to that of the okapi measure. At low ranks the performance was slightly superior: precision at 5 documents was 0.4560

for `cdS` and 0.4000 for `okS`. Nonetheless, preliminary experiments had suggested that the difference would be more pronounced. Local feedback based on the passages identified during cover density ranking improved performance, but once again our preliminary experiments had indicated that the improvement would be more pronounced. An initial examination of the detailed results indicates that the coordination level at which a query first matches has a influence on the retrieval performance; this observation appears to be related to the differences between our preliminary runs and our TREC-6 results.

As required, we submitted a run based on the topic descriptions (`uwmt6a2`). This run used a weighted variant of cover density to select passages for local feedback. Final document scoring was based on the MultiText implementation of the Okapi measure.

4.3 Manual Adhoc

For the manual adhoc task, we explored the effects of large amounts of interaction using shortest substring ranking and an interface that displayed relevant passages and allowed judgements to be recorded (Figure 5). No limit was placed on interaction time; four people spent a total of 105 hours over eight days (an average of 2.1 hours/topic) creating queries and making judgements for the 50 topics. It was our aim to use the high precision track and the manual adhoc task to explore the limits of the effects of user interaction — from 5 minutes (maximum) per query to 2 hours (average) per query. In addition, we wished to investigate the viability of this approach for creating a reasonable pool of judged documents with minimal effort.

The aim of the searchers was to find and judge as many relevant documents as possible. In judging the documents, the searchers placed them into three categories: relevant, not relevant, and borderline or "iffy". Queries were constructed manually to find potentially relevant documents, which were displayed in order of shortest substring ranking. In general, all documents were judged in this order until it was felt that further relevant documents were unlikely to be found. For most topics, several queries were issued to investigate different aspects of the topic.

We rendered 13,064 judgements in the process of completing the manual adhoc task. Agreement between our relevance judgements and those of the TREC judges was 77%, treating iffy as not relevant. However, for those documents judged to be relevant, agreement was relatively poor. Of the 3900 we found relevant, 2812 (72%) were officially judged: 1912 (68%) relevant and 900 (32%) not relevant. That is, only 49% of the documents we found and judged relevant were found and judged relevant using TREC-6 pooling and judging. Of the 4611 documents found relevant by the TREC judges, we judged 2699 (59%): we found 1912 (71%) relevant, 302 (11%) not relevant, and 485 (18%) iffy. Of the documents officially found relevant, we judged and found relevant 41% and judged and found iffy another 11%. These numbers are shown as a Venn diagram in figure 6.

Our final submission was derived from our judgements and from the queries constructed in the judging process. The judgements partitioned the documents into four categories: relevant, iffy, not relevant, and unjudged. The first tier of our submission was restricted to those documents judged relevant. Within this tier, documents were ranked using a manually created query and shortest substring ranking. The second and subsequent tiers were restricted to some combination of iffy, not relevant, and unjudged documents, selected subjectively by the searchers. For some topics, iffy or not relevant documents were deemed to be more probably relevant than unjudged documents; for others, the opposite. For example, for topic 301 ("International Organized Crime") we felt there remained many unjudged relevant documents and these were ranked before those judged iffy. The norm, however, was to rank iffy documents before unjudged ones, and to rank not relevant documents after unjudged ones.

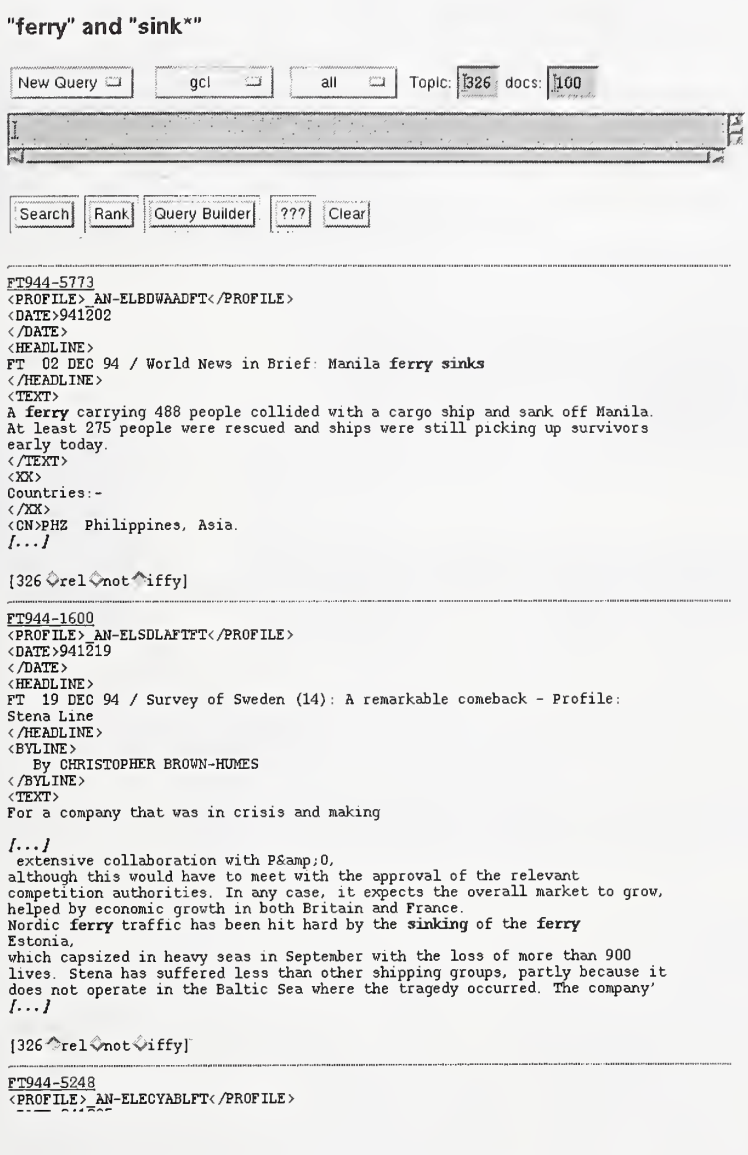


Figure 5: User Interface for Manual Adhoc Task

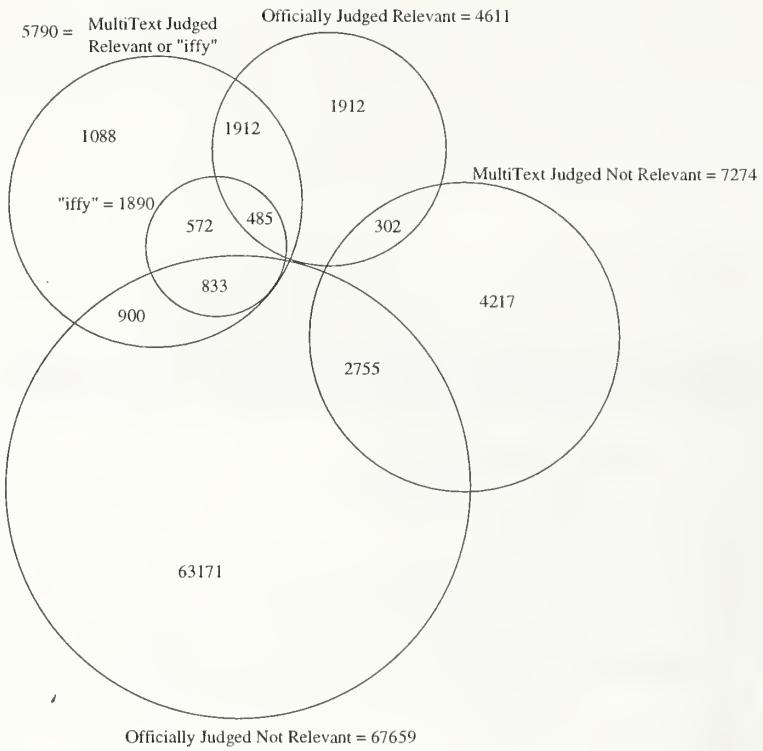


Figure 6: Judging Comparision

We examined user time, relevant documents, average precision, best average precision, and median average precision for all the topics. In general, large interaction times were correlated with large numbers of relevant documents, but not always correlated with precision. In particular, two topics for which high effort was expended and low precision results were achieved were topics 319 ("New Fuel Sources") and 301 ("International Organized Crime"). We attribute these particular results to misinterpretation of the topics by our searchers.

4.4 High Precision

For the high precision track, searchers attempted to find up to ten relevant documents within a fixed time period (five minutes). Any type of interaction was permitted. The MultiText high precision user interface is shown in Figure 7. The interface is based on "faceted" boolean queries. Each facet consists of a disjunction of terms; a query consists of a conjunction of facets.

A search begins when the participant selects the appropriate topic number. The five minute interval officially commences, the text of the topic is displayed, and the title of the topic is transformed into the initial query. The title is parsed into words and filtered against a list of stop words. The remaining words become distinct facets. Faceted queries are entered in the *Query Generator* window. Each line contains a facet and may contain any number of terms. Each term may be a word, a quoted phrase or even an arbitrary GCL query.

Once the query has been prepared, it is used to partition the documents in the database according to the maximal subset of the facets that occur in the document. For example, with facets A, B and C, we might partition the documents into the 8 possible sets ABC, $AB\bar{C}$, $A\bar{B}C$, etc. While

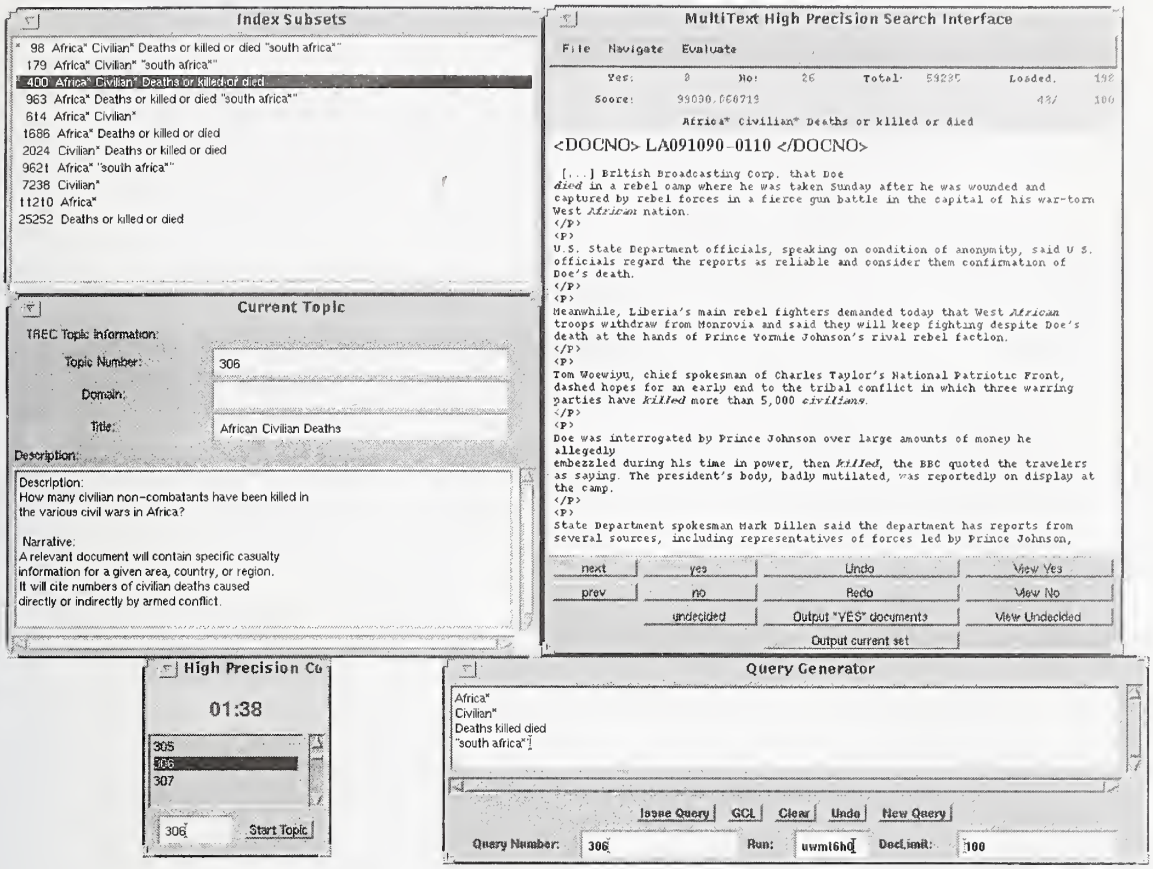


Figure 7: User Interface for High Precision Track

this could create an exponential number of sets, by using the maximal subset of the facets, in our experience, the number of subsets doesn't become unmanageable until there are approximately ten facets.

To summarize this partition, the *Index Subsets* window lists the cardinality of each set of documents and lists the facets present in the subset. For example, in figure 7 we see that there are 179 documents that contain "`africa*`", "`civilian*`", at least one of "`deaths`", "`killed`" or "`died`" but do not contain "`south africa*`". The list of subsets is sorted in descending order by the number of facets and then by the cardinality of the corresponding set of documents (in ascending order). This ordering tends to cluster the more promising sets at the top of the list.

Each entry in the subset list corresponds to a set of documents. The sets of documents may be individually navigated. Each set is ranked and displayed in order. Instead of displaying the entire document, the system displays relevant passages. The selected passages are those that contributed most to the score assigned to the document by the ranking function. The search terms in the facets will be displayed in a different color and typeface to facilitate locating the most relevant information in the passages. It is also possible to look at the entire document with the search terms in an alternate type and color. This feature was rarely used in the experiments. The passages were usually sufficient to determine a document's relevance.

Combining the subset construction and the ability to quickly read passages suggests a query

Run	Prec. @ 10	Rel. Prec @ 10	Avg Prec @ 10
uwmt6h0	0.5720	0.5977	0.0902
uwmt6h1	0.5680	0.5834	0.0982
uwmt6h2	0.5640	0.5951	0.0997
avg of best	0.7660	0.8084	0.1718
manual adhoc	0.6820		
avg of median	0.5340	0.5507	0.0782

Figure 8: High Precision Results

	Baseline (2GB)	Full (20GB)	Factor
Build Time (mins)	25.2	268.8	10.66
Query Time (secs)	11.34	67.3	5.93
Query Time (no overhead)	5.42	53.6	9.9
Time per topic (secs)	0.23	1.35	
Precision @ 20	0.498	0.643	

Figure 9: Very Large Corpus Results

refinement strategy. Starting from any given query, the user typically finds words that either strengthen or weaken the passage's likelihood of being relevant. For example, "shelling" may occur in many relevant passages and the user can then construct subsets that contain this word. Alternatively, passages about South Africa tend to be about police actions and therefore may not be relevant. By splitting the subsets on the phrase `"south africa*"`, the user can separate the subsets not containing South Africa from those that do. This refinement process is an interactive, manual version of the re-tiering used in the routing task. The searcher uses knowledge about the terms and information observed in the passages to build a mental model that resembles the relevance graphs described in Section 4.1

The table of Figure 8 compares the three submissions to the average of the best result for each topic and the median result for each topic. All runs were above the median. At least one of the three runs achieved the best precision @ 10 for 33 of the topics. At least two of the three runs tied with the best precision @ 10 for 12 of the topics. However, there is a high variance between topics; the average precision is 25% below the average of the best results. This indicates that certain users have problems with certain topics. It may be appropriate to work at integrating approaches used in the automatic adhoc to attempt to assist users when they do encounter difficulties.

4.5 Very Large Corpus

The MultiText system was designed as a distributed, scalable text database system. The Very Large Corpus track allowed experiments in software scalability and demonstrated the viability of the distributed architecture. The 2GB and 20GB collections of data were distributed over four inexpensive PCs. Queries were generated manually by interacting over the adhoc data. The official run measured software scalability and post-hoc experiments validated hardware scalability. The table of Figure 9 shows our VLC results.

To ensure that the baseline and full collections runs were comparable, the data was partitioned over the four PCs in an identical fashion. The four machines used a Cyrix processor (Pentium clone), three 3.8GB EIDE hard-drives, 64MB memory and consisted of inexpensive components.

No. Machines	1 (5GB)	2 (10GB)	3 (15GB)	4 (20GB)
Query Time (secs)	62.1	63.3	64.2	64.7
% Increase		1.9%	1.4%	0.8%
% Increase over 1 node		1.9%	3.4%	4.1%

Figure 10: Hardware Scalability

The machines ran a version of the Linux operating system. Most of the available storage was actually used. One drive held the system files as well as the original data. One drive held the index for 5GBs of data. The last drive held the document numbers for the 5GBs of data as well as the index and document numbers for the baseline data (500MBs). The remaining storage was used as temporary space during the original build.

Distributing the data over multiple nodes allows for parallel query processing. Since there are no collection statistics and every document is independently ranked, each processor can act independently. A simple marshaller/dispatcher can be used to dispatch a query to each of the machines and to collect the results by merging the lists of ranked documents. The marshaller/dispatcher can then contact the appropriate text databases in parallel to locate the document identifiers to provide the final answer to the query. The marshaller/dispatcher can be scaled to any number of nodes with a small per-machine overhead. The table of Figure 10 shows that there is only a moderate increase in time as nodes are added to the distributed database.

Database creation trivially scales to any number of nodes because each node builds a normal database. That is, each node will contain a valid database that may be accessed independently of the entire collection. The software scalability for the build is linear. Fixed sized hash tables and buffers are used to build partial indices that are written out to disk as the available memory is exhausted. The final step of the build is to merge the partial indices into the final index. Since I/O costs dominate, the total build time would be expected to scale approximately linearly with the size of the data. This expectation was observed in practice, with a 10.66 times increase in the build time when 10 times more data was added.

The queries were generated by interacting over the adhoc data using the adhoc queries as our basis. Short and precise queries were created. The queries averaged approximately 5.5 terms per topic. To improve performance, length restrictions were added to the queries to discard spurious solutions. Based on the probability estimations used in the routing task, discarding these solutions was not expected to affect retrieval effectiveness. The queries ran for an average of 1.35 seconds/topic. The GCL query algebra can be implemented in time that is at most linear in the length of the postings lists. Consequently, we expected a linear increase in time from the baseline to the full run. In the final analysis, the time taken to retrieve the document numbers was a significant overhead. It was this overhead that led to a sub-linear time increase. If we consider only the query processing time, we see the expected factor of 9.9 times.

There was increased precision in the full run over the baseline run. This increase is evidence that the ranking function is stable. Given more data, it is expected that a ranking method should find more relevant documents within a given rank. This expectation is consistent with the routing relevance graphs. Since there are more documents with higher scores, there are more documents that have a higher probability of relevance.

4.6 Chinese

This is the first year that we participated in the Chinese track. Our main goal was to evaluate how our approaches to ranking documents will perform in Chinese information retrieval environments.

As there are no explicit word boundaries in Chinese text, a decision has to be made on how to index the documents. We decided to index the collection as individual characters. To search for a word, we relied on the phrase searching capability provided by the MultiText retrieval system and looked for a sequence of adjacent characters. This post-coordination approach is well supported by our retrieval system and we believe it is more flexible than indexing segmented text. We can also avoid many of the problems with segmentation, such as the need for a large dictionary, the potential for erroneous character groupings, and the difficulty in handling new terms and proper nouns [10].

The queries for each topic were constructed manually in an interactive manner by one member of our research group. For each topic, an initial query was formulated and submitted to the retrieval system, and the top 10-15 documents returned were evaluated manually for their relevance. Additional terms that appeared salient in the relevant documents were added to the query, and terms that did not seem to be useful were eliminated. This process was iterated 2-3 times before the final version of the query was constructed.

For most topics, instead of using a single query that encompass the different aspects of the topic, a series of queries were formulated with each query focusing on a particular aspect of the topic. For example, several queries were constructed for topic 49: one focused on the extension of the non-proliferation treaty on nuclear arms, one focused on underground nuclear tests, and another focused on the promotion of peaceful use of nuclear power. If the queries all focused on important aspects of the topic they were combined disjunctively in a single tier. If some queries were considered more important than others, tiered ranking was used to put these queries ahead of the more peripheral ones. The queries for each topic took about 1.5-2 hours to formulate. The final queries would be suitable for use in a future routing task.

The results of our only run, `uwmt6c0`, were encouraging. We were above the median in average precision in 21 of the 26 topics, while finishing top in 7 of them. The results show that our ranking approach performs well for Chinese document collections.

References

[1] Charles L. A. Clarke and Gordon V. Cormack. Interactive substring retriveal. In *Fifth Text REtrieval Conference (TREC-4)*, 1996.

[2] Charles L. A. Clarke, Gordon V. Cormack, and Forbes J. Burkowski. Schema-independent retrieval from hetrogeneous structured text. In *Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 279–289, Las Vegas, Nevada, April 1995.

[3] Charles L. A. Clarke, Gordon V. Cormack, and Forbes J. Burkowski. Shortest substring ranking. In *Fourth Text REtrieval Conference (TREC-4)*, pages 295–304, Gaithersburg, Maryland, November 1995.

[4] Charles L. A. Clarke, Gordon V. Cormack, and Elizabeth A. Tudhope. Relevance ranking for one to three term queries. In *Fifth RIAO Conference*, pages 388–400, Montreal, June 1997.

[5] W. Bruce Croft, Robert Cook, and Dean Wilder. Providing government information on the Internet: Experiences with THOMAS. In *Digital Libaries Conference (DL'95)*, pages 19–24, Austin, Texas, June 1995.

[6] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Possion method for probabalistic weighted retrieval. In *17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 311–317, Dublin, July 1994.

[7] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Third Text REtrieval Conference (TREC-4)*, pages 109–126, 1994.

[8] Daniel E. Rose and Curt Stevens. V-Twin: A lightweight engine for interactive use. In *Fifth Text REtrieval Conference (TREC-5)*, 1996.

[9] Ross Wilkinson, Justin Zobel, and Ron Sacks-Davis. Similarity measures for short queries. In *Fourth Text REtrieval Conference (TREC-4)*, pages 277–285, 1995.

[10] Zimin Wu and Gwyneth Tseng. Chinese text segmentation for text retrieval: Achievements and problems. *Journal of the American Society for Information Science*, 44(9):532–542, 1993.

Mercure at trec6

M. Boughanem [1 2] **C. Soulé-Dupuy** [2 3]

[1] MSI
Université de Limoges
123, Av. Albert Thomas
F-87060 Limoges

[2] IRIT/SIG
Campus Univ. Toulouse III
118, Route de Narbonne
F-31062 Toulouse

[3] CERISS
Université Toulouse I
Manufacture des Tabacs
F-31000 Toulouse

Email : {bougha, soule}@irit.fr

1 Introduction

We continue our work in trec performing runs in adhoc, routing and part of the cross language track. The major investigations this year are the weight schemes modification to take into account the document length. We also experiment the high precision procedure in automatic adhoc environment by tuning the term weight parameters.

2 Mercure model

Mercure is an information retrieval system based on a connexionist approach and modelled by a network (as shown in the figure 1) containing an input representing the query, a term layer representing the indexing terms, a document layer representing the documents and an output representing the retrieved documents. The term nodes (or neurons) are connected to the document nodes (or neurons) by weighted indexing links. Mercure includes the implementation of two main components : the query evaluation based on spreading activation from the input to the output through the indexing links and the automatic query modification based on backpropagation of the document relevance.

2.1 Query evaluation based on spreading activation

The query evaluation is performed as follows :

1. Build the input $Input_k = (q_{1k}, q_{2k}, ..., q_{Tk})$,
2. Apply this input to the term layer. Each term neuron computes an input value :

   $$In(N_{t_i}) = q_{ik}$$

   and then an output value : $Out(N_{t_i}) = g(In(N_{t_i}))$
3. These signals are propagated forwards through the network. Each neuron computes an input and an output value :

   $$In(N_{D_i}) = \sum_{j=1}^{T} Out(N_j) * w_{ij}$$

   then, $Out(N_{D_i}) = g(In(N_{D_i}))$

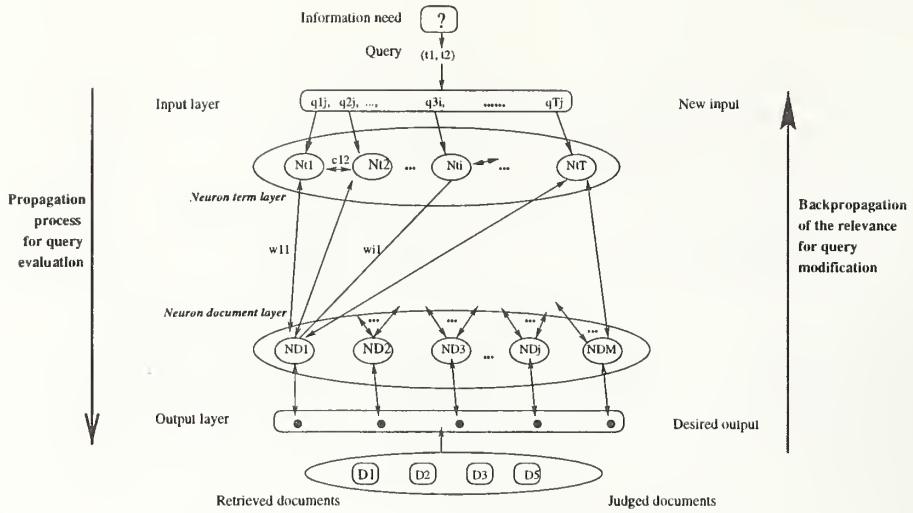


Figure 1: *The Mercure Model.*

The output vector is : $Output_k(Out(N_{D_1}), Out(N_{D_2}), ..., Out(N_{D_M}))$

These output values computed by the document neurons are used to rank the list of retrieved documents.

2.2 Query modification based on relevance backpropagation

The automatic query modification is based on spreading the document relevance values backwards the network. The retrieved documents are used to build the *DesiredOutput*. To each judged document is assigned a *relevance value*. A positive relevance value is assigned to relevant documents, a negative value to non-relevant documents. The desired output is represented by the vector of the form : $DesiredOutput = (rel_1, ..., rel_i, ..., rel_M)$.

This strategy consists in backpropagating the relevance values from the output layer to the input layer, and it is performed as follows :

1. Build the desired output : $DesiredOutput = (rel_1, ..., rel_i, ..., rel_M)$,
2. Apply this output to the neuron document layer. Each neuron computes an input value :

$$In(N_{D_i}) = rel_i$$

and then an ouput signal : $Out(N_{D_i}) = g(In(N_{D_i}))$

3. The output signals are backpropagated to the term neuron layer. Each neuron term computes an input value :

$$In(N_{t_i}) = \sum_{j=1}^{M} (w_{ij} * Out(N_{D_j}))$$

and then an output signal : $Out(N_{t_i}) = g(In(N_{t_i}))$

4. A new input is then computed according to this formula :

$$NewInput_k = \alpha * Input_k + \beta * Out(N_t)$$

This new input is applied to the term neuron layer and a new query evaluation is then done.

Several formulations can be used to construct the desired output. For this experimentation we have chosen the following formula :

- *for relevant document* : $rel_i = \frac{Coef\_Rel}{Nb\_rel}$
- *for nonrelevant document* : $rel_i = \frac{Coef\_NRel}{Nb\_Nrel}$

Where :
$Coef\_Rel$, $Coef\_NRel$: relevance coefficient of the documents (positive for relevant and negative for non-relevant documents),
Nb_rel, Nb_Nrel : number of relevant and non-relevant documents respectively,

3 General Investigations

Our first investigation is to modify the indexing weight to take into account the document length. Our formula is inspired by Okapi and Smart term weight functions. It is expressed by :

$$w_{ij} = \frac{\frac{(1+log(tf_{ij}))}{1+log(average_j(tf_{ij}))} * (h_1 + h_2 * log(\frac{N}{n_i}))}{h_3 + h_4 * \frac{doclen_j}{avg\_doclen}}$$

The query term weight in the input is expressed by :

$$q_{ik} = \frac{(1 + log(tf_{ik})) * (log(N/n_i)}{\sqrt{\sum_{j=1}^{T} (1 + log(tf_{jk})) * (log(N/n_j)^2}}$$

Where :
w_{ij} : the weight of the link between the term t_i and the document D_j,
tf_{ij} : the frequency of the term t_i in the document D_j,
T : the number of documents in the collection,
n_i : the number of documents containing the term t_i,
$doclen_j$: document length in words (without stop words),
avg_doclen : average document length, computed for each database.

4 Adhoc experiment and results

4.1 adhoc methodology

Our investigation is to improve the query expansion in automatic adhoc environment. The "blind" relevance feedback was performed by assuming the top retrieved documents as relevant and the low retrieved as non relevant. Some efforts have been undertaken to improve the precision in the small top ranked documents. The basic goal is to produce the "High precision" by "trading" the recall for the precision, [4] [5] (e.g we can loose some relevant documents if we are sure that the remaining ones are relevant).

A way to produce a high precision could be by using "good" query term and document term weights. Our strategy in adhoc trec-6 is to weight the indexing links in order to maximize the precision at small ranked top documents and then a "normal" weight scheme (weight performing a best precision at 1000 top ranked documents) will be used in the relevance backpropagation process and in the new input spreading. The weight schemes we used in trec-6 are obtained by tuning the h_1, h_2, h_3, h_4 parameters.

Series of experiments have been undertaken on TREC-5 database and queries. The parameters we have chosen to use in TREC-6 experiment are : $h_1 = 1$, $h_2 = 0$, $h_3 = .8$, $h_4 = .2$ for the high precision and $h_1 = .8$, $h_2 = .2$, $h_3 = .8$, $h_4 = .2$ for what we called a "normal" weight. The remaining parameters used in the relevance backpropagation are : $Coef\_Rel = 1$, $Coef\_NRel = -.75$, $\alpha = 2$, $\beta = .5$, $Nb\_rel = 12$, $Nb\_Nrel = 500$ (from 501 to 1000).

4.2 Adhoc results and discussion

Preliminary investigations

The first result we underline concerns the term weight functions. The table 1 shows the average precision of basic run obtained by some IR systems in TREC-5. We can notice that the weight schemes we used are quite good ($h_1 = .8$, $h_2 = .2$, $h_3 = .8$, $h_4 = .2$).

TREC-5 results	
system	average precision in initial search
Mercure :	0.1578
Okapi :	0.1520
Smart :	0.1484
Inquery :	0.1442

Table 1: Comparative basic search trec-5 results

Automatic adhoc results

Three automatic runs were submitted : Mercure2 (description only), Mercure1 (long topic : title, description and narrative) and Mercure3 (title only). These runs were based on completely automatic processing of TREC queries and automatic query expansion, the high precision concept was also used. Table 2 compares our runs against the published median runs. We notice that most of the runs are above the median.

TREC results			
Run	Best	$\geq$median	$<$ *median*
Mercure2 (description)	1	40	10
Mercure3 (title)	1	29	18
Mercure1 (long topic)	5	44	6

Table 2: Comparative automatic adhoc results at average precision

We unfortunatelly noticed an error in the script that has been used to perform the adhoc description run (the other runs are right). The weight scheme (i.e. the h_i parameters) used to produce the high precision has also been used by mistake in the relevance backpropagation process instead of the "normal" h_i values. The table 3 shows the official and the corrected runs).

We actually notice a difference between the description runs, the other runs seem good.

The table 4 the average precisions of the basic run using the high precision and the run after query expansion on the three corrected runs. The query expansion is done by using the following values of Mercure parameters : $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relevant documents and the number

Run	Official results		corrected results	
	Average precision	R. Precision	Average precision	R. Precision
Mercure2 (description)	0.1640	0.2065	0.1720	0.2108
Mercure3 (title)	0.2316	0.2689	0.2316	0.2689
Mercure1 (long topic)	0.2305	0.2700	0.2305	0.2700

Table 3: Automatic adhoc results - 50 queries

pf term added to the query is 16. We notice that the automatic query expansion is still effective in the adhoc environment.

Run	average precision
Mercure3 : title only	
basic search using h_i producing the high precision	0.2041
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.2316 (+13.47 %)
(Mercure2.C) description only	
basic search using h_i producing the high precision	0.1549
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.1710 (+10.39 %)
(Mercure1) long topic	
basic search using h_i producing the high precision	0.2128
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.2305 (+8.32 %)

Table 4: Adhoc component results - 50 queries

However we notice that the way used to improve the precision at top ranked documents did not have a positive effect as in the trec-5 adhoc . Indeed, the table 5 shows the results in the description run (Mercure2.C.N) when using the "normal" h_i values. We observe a slight different in favour of the Mercure2.C.N run. We do not yet analyze the results of the title and long topics runs.

Run	average precision
Mercure2.C.N: description only	
basic search using "normal" h_i	0.1693
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-100 non-rel docs	0.1772

Table 5: Adhoc component results - 50 queries

5 Routing experiment and results

All trec-6 training data were used (relevant and non relevant documents). The queries are initially built automatically from all the fields of the topics and then expanded by using the 30 top terms resulting from the relevance backpropagation procedure.

Each query was evaluated by varying the different Mercure parameters, h_i and α, β, etc. The queries performing the best average precision in the training data were selected. Moreover, a slight modification has been performed in the relevance value formula, it concerns the positive relevance value. Indeed, we decided to take into account the fact that a relevant document is or not among the 1000 retrieved documents in the initial search.

The relevance value assigned to each relevant document becomes :

- $rel_i = \frac{coef\_R}{Nb\_rel} * BOOT \begin{cases} BOOT = 1 \; if \; the \; relevant \; document \; is \; not \; in \; the \; 1000 \; documents \\ BOOT < 1 \; if \; relevant \; document \; is \; retrieved \; (BOOT = .9 \; for \; routing \; trec6) \end{cases}$
- no modification if a document is nonrelevant

As the retrieved relevant documents are already close to the initial query, we give to the terms occurring in the non retrieved relevant documents more effect in the final query building.

The table 6 compares our routing runs against the medians published runs, more than 60% of queries are above the median.

TREC routing results			
Run	Best	$\geq$median	$<$ *median*
Mercure4	1	29	18

Table 6: Comparative TREC Results at average precision

The table 7 shows the difference between the run based on the initial queries and the one based on the routing queries. We ha have no time to analyze these results

TREC routing results			
Run	average precision	R precision	Total Rel retrieved
Mercure4	0.3061	0.3400	4774

Table 7: Comparative TREC Results at average precision

Run	average precision
basic search (with the initial queries)	0.2676
Official run	0.3061

Table 8: Routing component results 47-queries

6 Cross language track : french to french

Two runs french to french were submitted in CLIR track. The indexing and search methodologies are the same than the adhoc trec6 except the stemming algorithm where a cutoff stemming method (7 characters) has been used. This stemming method has been implemented in all of our operational information retrieval systems dealing with french documents and french queries. The results obtained untill now lead us to go on the experiments with this stemming method.

Moreover, for the adhoc task the high precision procedure has not been used because there is no relevance information to tune the weight scheme. The same parameters were used for the indexing weight $h_1 = .8$, $h_2 = .2$, $h_3 = .8$, $h_4 = .2$.

The table 9 compares our runs against the published median runs. Most of the queries are above the median.

TREC-6 cross language french to french			
Run	Best	$\geq$median	$<$ *median*
MercureFFs (description)	0	18	3
MercureFFl (long topic)	4	17	4

Table 9: Comparative TREC cross language at average precision

The table 10 shows that the average precision and the R-precision for the different runs are quite good.

Run	Average precision	R. Precision	Total Rel Retrieved
MercureFFs (description)	0.3619	0.3848	1023
MercureFFl (long topic)	0.3778	0.4015	1033

Table 10: cross language (french to french) results - 21 queries

The important point we discuss concerns the automatic query expansion. Indeed, the table 11 shows the improvment obtained between the basic run and the run with an automatic query expansion using the following values of Mercure parameters : $Nb\_rel = 15$, $Nb\_Nrel = 500$, 501-1000 non-relevant docs and the number of added terms is 16. In both, MercureFFs and MercureFFl the improvement about 10%.

Run	average precision
description only	
basic search	0.3262
Expansion $Nb\_rel = 15$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.3619 (11%)
long topic	
basic search	0.3479
Expansion $Nb\_rel = 15$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.3778 (8.6 %)

Table 11: Adhoc cross language component results - 21 queries

7 Conclusion

Last year, we participated in trec-5 in the adhoc and routing tasks in category B. Our main effort this year has been to participate in trec-6 in category A. We performed completely automatic runs in adhoc, routing and a part of the cross language tasks.

At first we planed to try, the passage retrieval, the data mining techniques [7] and the genetic algorithms [1] to automatically expand the queries. But finally, our investigations were the improvement of the term weighting and the automatic query modification. We spent much time on these experiments and decided to difer the planed experiments until the next year.

However, the results we obtained for the main tasks are still encouraging this year. Our participation to the CLIR track was limited to a french to french experimentation to train our french language processing. Our goal now is to go on with a real cross language experiment.

References

[1] L. Tamine Reformulation de requêtes basée sur l'algorithmique génétique Proceedings of INFORSID'97 Toulouse Juin 1997.

[2] M. Boughanem & C. Soule-Dupuy, *Query modification based on relevance backpropagation*, Proceedings of the 5th International Conference on computer-assisted information searching on Internet (RIAO'97), Montreal, June 1997.

[3] M. Boughanem & C. Soule-Dupuy, *Mercure : adhoc and routing tasks*, 5th International Conference on Text REtrieval TREC2, Harman D.K. (Ed.), NIST SP 500-236, 1996.

[4] C. Buckley & al, *Query zoning : TREC'5*, 5th International Conference on Text REtrieval TREC2, Harman D.K. (Ed.), NIST SP 500-236, 1996.

[5] B. croft, & al, INQUERY at TREC-5. 5th International Conference on Text REtrieval TREC5, Harman D.K. (Ed.), 1996.

[6] S. Robertson and al, *Okapi at TREC-5* . 5th International Conference on Text REtrieval TREC2, Harman D.K. (Ed.), NIST SP 500-236, 1996.

[7] T. Dkaki, B. Dousset & M. Mothe, *Mining information in order to extract hidden and strategical information*, Proceedings of the 5th International Conference on computer-assisted information searching on Internet (RIAO'97), Montreal, June 1997.

Daimler Benz Research:
System and Experiments
Routing and Filtering

Thomas Bayer, Heike Mogg-Schneider, Ingrid Renz, Hartmut Schäfer
Daimler-Benz AG
Research and Technology
Wilhelm Runge Str. 11
D - 89081 Ulm
Germany
email: renz@dbag.ulm.daimlerbenz.com

1 General Approach

The retrieval approach is based on vector representation (bag of character strings), on dimension reduction (LSI - latent semantic indexing) and on statistical machine learning techniques in all processing levels. Two phases are distinguished, the *adaptation* phase based on training samples (texts) and the *application* phase, where each text is mapped to one or more categories (classes). The adaptation process is corpus dependent and automatic and, hence, domain and language independent.

The main idea of this approach is to generate different sets of simple features which represent different views to texts. For each text to be filtered/routed, different feature vectors are generated and classified into a decision vector which contains estimates of class membership probabilities. In the following step, these decision vectors are regarded as feature vectors and fed to another classifier that combines these set of decision vectors into the final one.

2 System Overview

Fig. 1 illustrates the principle design of the system (for more details about our work see [1] and [2]). First, resources are adapted using a text (training) corpus and deferred measurements of this corpus (steps (1) - (5)), such as feature and decision vectors; in these steps the algorithms described above are applied. After adaptation of all resources, the system is able to assign one or more categories along with probabilities to an unknown text.

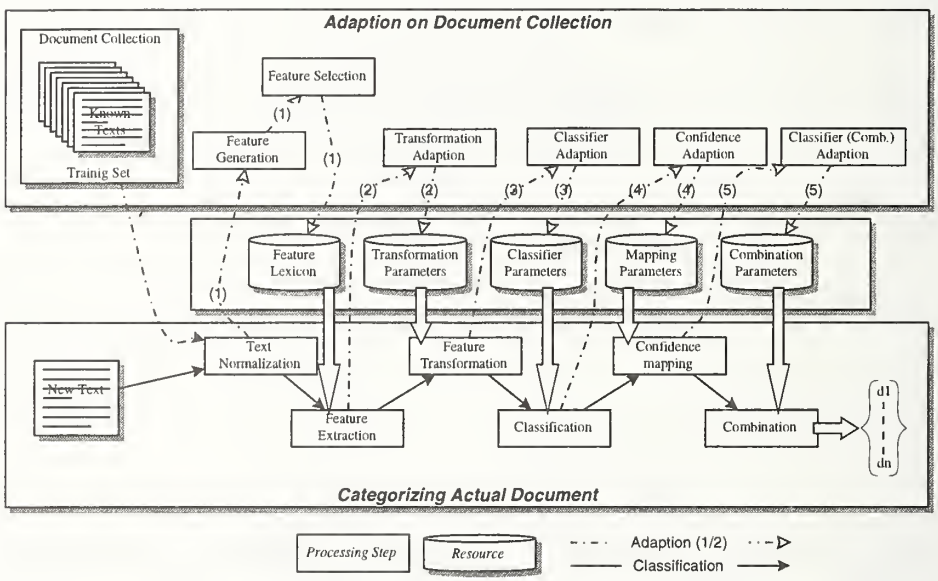


Figure 1: Adaptation and application are distinguished.

The final decision is represented as a decision vector. In case of routing the estimated probabilities d_k of each category k are sorted. In case of filtering – here, a binary decision has to be made – thresholds are calculated for each category; if the estimated probability is above the threshold, this category is assigned to the text, if it is below, the text is rejected with respect to this category.

We regard the TREC task as a 47-class problem, and therefore, construct classifiers with the ability to distinguish between 47 classes. An alternative approach would have been to construct 47 different 2-class classifiers

each being responsible for one class and the complement of this class. However, this approach would have multiplied the computational effort in the optimization phase.

The systems for routing and filtering do not differ with respect to the processing steps except the final step of threshold comparison in case of filtering. However, the training samples and the number of samples for both tasks differed significantly. For both systems confidence mapping (see fig. 1 is not used because it did not significantly improve system's accuracy during evaluation.

3 Processing Steps

3.1 Textnormalization

The text is converted to a normalized form: all HTML tags are removed; all numbers converted to the digit 1; each word is converted to lower case; punctuation is removed. Each white-space character is replaced by the blank character.

3.2 Feature Generation

With two procedures (simple n-gram statistics and corpus-based sub-string computation), we computed different feature sets which represent different views to these texts. In every feature set, a feature is a string, i.e. a sequence of 3 or 4 characters (alphanumeric, blank), - this limitation of length is due to efficiency of text transformation. Every feature set only results from the normalized and complete texts of the training set. Neither topic descriptions nor external knowledge bases (thesauri, lexica) are used, but the information to which topic a text belongs.

N-gram statistics

For every topic, all word forms of the training texts are transformed into topic-specific lists of 3- and 4-grams together with their frequencies. Only frequent and topic-specific n-grams are selected as features.

For routing, we selected a set of 4242 3-grams and 5496 4-grams. For filtering, we chose 3820 3-grams and a set of 7596 4-grams.

Corpus-based sub-string computation

This approach acquires features by breaking words of the training set into parts (sub-strings) by an iterative method and collects them into a lexicon.

First, stop-words are defined as word forms which are very frequent and equally distributed among all topics. Then, affixes are computed as frequent beginnings and endings of word forms. Word forms which are not stop-words are split into smaller parts exploiting the morphological regularity that complex words are composed of simpler forms. If a complex form and one of its components are both members of the list of word forms, the complex string is divided into the component and the remaining character string.

The resulting sub-strings are ranked according to different selection criteria: chi-square, tf/idf-measure, correlation measure. For every topic, the first 50 sub-strings of these rankings are collected as features and transformed into 4-grams which are more tractable in the following step of text transformation.

The number of features in these sets are 831, 1995, and 4369 for routing and 2288, 3368, and 4146 for filtering.

Hence, we generated 5 feature sets for routing and 5 for filtering. They are different because the training texts have been different. For routing the dimension L ranges from 831 to 5496, for filtering from 2288 to 7596.

3.3 Text Transformation

According to the five feature sets, every text is transformed into five vectors. For every feature, the text frequency is computed and inserted into an indexed vector v_0.

3.4 Dimension Reduction

The original dimension L is reduced to a small number L' of several hundreds. One well known method to reduce the vector space R^L is the principal component analysis (PCA) which is based on the eigenvalues and eigenvectors of the covariance matrix of the sample vectors.

The reason for dimension reduction is that the higher L is, the more training examples must be provided in order to avoid overfitting of the subsequent

classifier to the training set.

In Information Retrieval, SVD (singular value decomposition) is typically used instead of the PCA for dimension reduction (*latent semantic indexing*, see [3]): $Y = UDV$, where Y contains all feature vectors of the training set. It can be shown that the L eigenvectors of the covariance matrix are identical to the vectors of matrix U. However, PCA requires less computation both in terms of time and space.

Depending on the original dimension of the vector space and the number of training samples we selected the most efficient way with respect to computational effort: if the number of samples N is lower than the dimension of the feature space L – which is mostly true for this input data – the Gram matrix of the feature vectors (dimension $N \times N$) should be used; otherwise, the covariance matrix is more appropriate (dimension $L \times L$).

3.5 Classification

The problem of routing/filtering is regarded as a classification task into K categories (K-class problem). Hence, a classifier maps a feature vector into a K-dimensional decision vector where each component represents the a-posteriori probability that this feature vector belongs to category k. Another approach would have been to regard this problem as K different problems in order to distinguish one class k from all others. This approach would have led to K 2-class problems.

For classification into one of K categories (decision space), a numerical classification technique is employed. In [6] different numeric classifiers are compared, ranging from the standard Rocchio approach to linear and non-linear neural networks. The classification principle employed here is function approximation based on polynomials [5]. The L' elements of $v \in R^{L'}$ (derived from v_0 by PCA) are combined by a polynomial function $x : v \rightarrow x(v)$ resulting in multiplicative combination of the elements. Mathematically, the polynomial classifier is defined as $d = A^T \times x(v)$, where $A \in R^{K \times X}$ is the coefficient matrix to be adapted and X the dimension of the range of the function x. The coefficients are calculated by minimizing the mean-square error between the estimation d and the true value y – the target vector which is a unit vector with the 1 at the k-th position – describing the category membership of v:

$$E\{|A^T \times x(v) - y|^2\} = Minimum(A).$$

$E\{\ldots\}$ denotes the mathematical expectation In the optimization problem above A is computed by regression assessing a training sample of size N of pairs (v^i, y^i). It can be shown ([5]) that the k-th element of d estimates the a-posteriori probability $p(k|v)$. For a detailed description of the polynomial classifier design see [5] and [4].

The concept of the polynomial classifier has been extended with respect to y. y is a unit vector indicating that an object belongs to exactly one class. For categorization tasks this assumption does not hold any more; therefore, y contains as many 1's as class memberships exist. This includes some consequences for subsequent processing which are not discussed in detail here. However, the essential consequence is that the resulting decision vector can not be normalized to length 1.

The linear classifier is identical to the LLSF (linear least square fit) classifier described by Yang (see [7] and [8]). However, the mathematical principle is different in general if higher order polynomials are used. In this case, a non-linear function (e.g. quadratic polynomial) maps the feature space to the decision space yielding better separation of categories in the decision space.

4 Experiments

The relevant texts provided by NIST (24276 for routing, 4925 for filtering) have been divided into 75% training and 25% test texts for each of the 47 categories. With respect to fig. 1, the training texts represent the document collection from which all resources are derived. The full training text has been used for adaptation – typically everything between the HTML markers <TEXT> </TEXT> – although we noticed during evaluation that the relevant portion of a text may only be a fraction of the original text. For adaptation, only the training texts have been observed in all steps.

The test samples only served for optimization of parameter setting of each processing step. When accuracy scores are reported in the following, these scores have been obtained from runs on the test set.

All texts marked as relevant had been used for training and test. Non-

relevant texts (marked as non-relevant and not marked and thus presumed to be non-relevant) have not been considered for adaptation, but some of them (2000 marked as non-relevant and 3000 not marked) have been included into the test set in case of filtering in order to determine the thresholds for rejects for each category. Topic files have not been used.

In optimizing the system's accuracy for routing/filtering, we experimented with different parameter settings for all processing steps except *Text Normalization*. In routing, the optimization criteria were (non-)interpolated average precision (IAP and NIAP in the following) for the (a-priori probability weighted) mean value of IAP and NIAP for all 47 categories. In filtering, we used the mean value of ASP, F1, and F2 for all 47 categories (equally weighted).

Feature generation: Several parameters could have been varied: the selection of n-grams (what is frequent - the most frequent 20%, 50% of all n-gram, what is topic-specific - in 5%, 20% of all topics), the selection criteria of the corpus-based sub-strings (information gain, mutual information, signal-to-noise-ratio) or the norming of feature vectors (binarization, normalization with max. value or to 1). Because of time constraints, none of these variations have been evaluated in detail.

Dimension reduction: For each of the 5 feature lexica the transformation matrix had been calculated based on the feature vectors of the training texts (the class membership knowledge is not used here). Depending on the original dimension of the vector space and the number of training samples we selected the most efficient way with respect to computational effort (see sect. 2.3), PCA of either the covariance or the Gram matrix.

In routing, the original dimensions of the feature space ranged from 831 to 5496. We experimented with reduced dimensions of 300-600-900-1200 for each of the 5 feature sets. We selected a dimension of 900 to reduce the different features to because the evaluation on the test texts showed that the accuracy raised up to 10% (in mean for all 5 feature sets) from 300 to 900, but did not raise in case of 1200. Additionally, a vector space of dimension 1200 may lead to an overfitting of the classifier because the size of the training set is below 20000 samples. The highest score for dimension 900 was 80% IAP and 84% NIAP for feature routing-1995.

Since the size of the training corpus was much smaller in case of filtering, we experimented in the range of 300-450-600-750 to reduce the original di-

mension which was in the range of 2288 and 7596. We decided to reduce to a dimension of 600 because the F1 and F2 values measured derived from the test set were maximized in this case. We obtained the best F1/F2 scores for feature filtering-3820 of 13 and 9 respectively.

Generating the coefficient matrix for dimension reduction is computational expensive (time and particular space); but calculation must be done only once for a given feature set. We used a Pentium 200MHz computer with 512MB RAM. This machine enables us to calculate eigenvectors from a matrix of size 6000×6000 (either Gram or covariance matrix). Transformation during application is fast, however.

Classification: We experimented with linear and quadratic polynomial classifiers and with a distance measuring (euclidean) classifier where each category was represented by the mean-value of its training vectors. Since the feature vectors are normalized to length 1, this reference classifier is identical to the Rocchio classifier using the cosine measure.

The evaluation showed that the polynomial classifier constantly outperformed the Rocchio approach. Finally, we decided to select linear polynomial classifiers for each of the five feature sets (for filtering and routing) because for the second order polynomial the number of parameters to adapt (the matrix A) grows quadratic with the dimension of the feature space and therefore, one would need more training data than the ones having been available. Hence, the experiments had shown that this classifiers are overfitted and have not been able to generalize on the test texts.

In order to overcome the problem of dimensionality, we also tried to reduce the dimension by PCA to a range of 50. However, the loss of information is too high in this case, which could not been compensated by the more powerful second order classification approach.

Combination: Each text has been represented by 5 different feature vectors resulting in 5 different decision vectors. All or a subset of them can be concatenated to a single vector which is regarded as a new feature vector of dimension $m \times 47$. A linear polynomial classifier is adapted on the training set of concatenated vectors resulting in the final decision vector for a text.

We experimented with different subsets in routing and filtering. The best scores have been achieved by combining all 5 decision vectors. Even the worst local subsystem improved the accuracy of an arbitrary subset of the remaining ones after combination with them. Without combination, the

best local scores for IAP and NIAP have been 80.39% and 84.92% for the routing-1995 (LSI to 900) feature; combining all, the scores improved to 85.98% and 90.11% (note that the test texts only contained relevant texts). We detected the same behavior in case of filtering; we obtained the best local F1/F2 scores for feature filtering-3820 (LSI to 600) of 13 and 9 respectively. After combination they raised to 16 and 14 (in this case, 2/3 of the test text had been non-relevant ones).

Threshold selection: In case of filtering we selected thresholds for each category by maximizing the system's responses for the three different evaluation measures F1,F2 and ASP resulting in three 47-dimensional threshold vectors. Hence, this decision also has been adapted by the sample set. Since the F2 measure considers recall (penalizes missing relevant ones) the thresholds are lower than the F1 thresholds for all categories.

When applying the system, each decision vector of a text from the TREC-6 test set was compared against three different threshold vectors; the text was assigned to a category if the component of the decision vector exceeded the threshold component.

5 Results and Discussion

In the following some evaluations of our results for routing and filtering are presented as they were distributed from NIST.

5.1 Routing

The results for routing are summarized in the following two tables in fig. 2 which shows a comparison of our system with the other ones across the 47 topics; the left diagram represents the evaluation of the interpolated recall, the right one the precision at 1000 documents. The four bars in each diagram show how often the system fell to minimum, below median, above or equal median, or reached maximum.

The following tables show the results in detail:

```
Total number of documents over all queries
    Retrieved:    47000
    Relevant:      6872
```

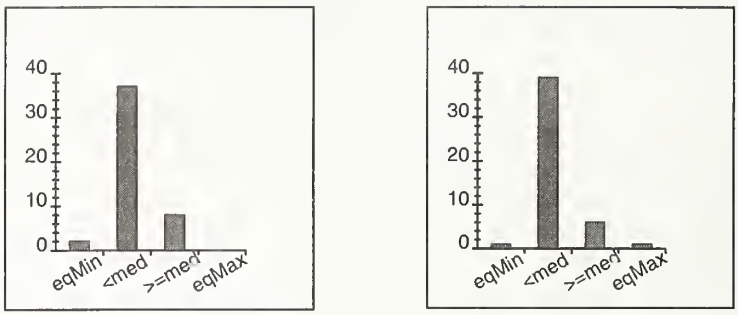


Figure 2: Comparison of our system with the other ones across the 47 topics; four values are displayed: the frequency the system achieved the minimum score, was below median, equal or above median and achieved maximum. The left one refers to interpolated recall, the right one to precision at 1000 documents.

```
    Rel_ret:       3334
Interpolated Recall - Precision Averages:
    at 0.00        0.6299
    at 0.10        0.4131
    at 0.20        0.3027
    at 0.30        0.2240
    at 0.40        0.1718
    at 0.50        0.1222
    at 0.60        0.0845
    at 0.70        0.0620
    at 0.80        0.0330
    at 0.90        0.0073
    at 1.00        0.0029
Average precision (non-interpolated) over all rel docs
                   0.1619
Precision:
  At    5 docs:    0.4213
  At   10 docs:    0.3830
  At   15 docs:    0.3645
  At   20 docs:    0.3553
  At   30 docs:    0.3326
```

```
  At  100 docs:   0.2436
  At  200 docs:   0.1820
  At  500 docs:   0.1102
  At 1000 docs:   0.0709
R-Precision (precision after R (= num_rel for a query)
docs retrieved):
    Exact:        0.2199
```

The results are disappointing. Three factors seem to be responsible for these unsatisfactory results:

1. The system had been trained with all relevant texts of the training set which comprises texts from quite different information sources, like AP, Wall Street Journal, etc. However, these texts are not representative for the test texts which come from the FBIS information source entirely. Another clue to this fact is that our filtering system which was trained with FBIS texts only and has a similar architecture was able to retrieve 3804 relevant texts compared to 3334 of the routing system. As a consequence the features trained do not necessarily match the ones of the FBIS texts.

2. We used all the text for training a text sample contained although we noticed during evaluation that only a small fraction of a text may really be relevant. Therefore, the feature space is further moved to a non-representative one.

3. The routing system was – mistakenly – evaluated by averaging the 47 topics due to their a-priori probability instead of averaging equally weighted (as NIST does in their evaluations). This affected the PCA and the classifier. Therefore, the optimum performance may not be reached.

5.2 Filtering

The filtering results are more encouraging as the figures for the evaluation measures show in fig. 3. The pooled results show the performance of the submitted results, the ranked bars show the potential improvement if optimum thresholds would be selected.

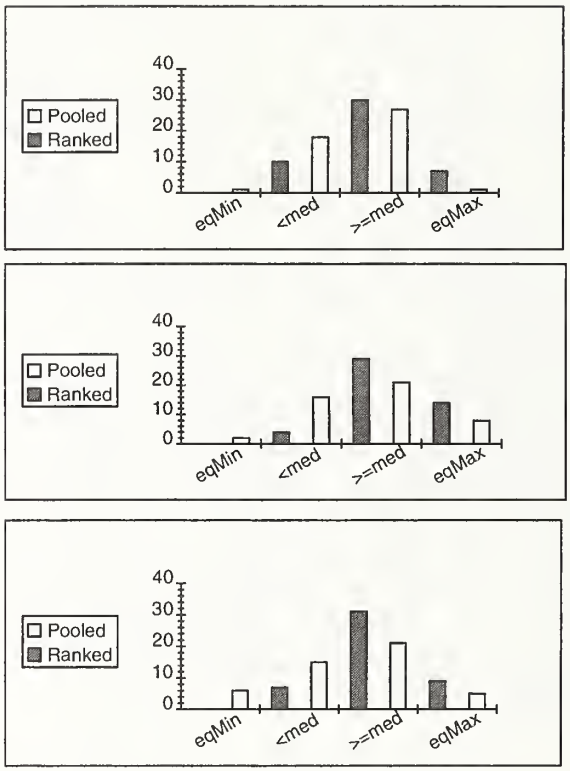


Figure 3: Comparison of our system with the other ones across the 47 topics; the same four values are displayed as in the previous figure. ASP, F1 and F2 are shown from top to bottom.

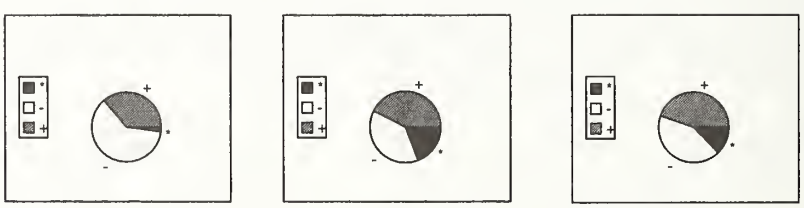


Figure 4: Evaluation of thresholds for ASP, F1 and F2. '+' means too many texts have been selected, '-' too few and '*' optimum.

The pooled scores of ASP, F1, and F2 are good: clearly, the scores lie above median. ASP and F1 are slightly better than F2; the reason is that the thresholds are smaller for F2 and hence, more documents are retrieved which are not relevant and therefore degrade the system's performance.

The ranked results in fig. 3 and the charts in fig. 4 both evaluate the threshold selection algorithm. The comparison between ranked results and pooled results shows that the thresholds are not optimum but reasonably good for F1 and F2. In these cases, roughly 20% of the thresholds are optimum and the remaining ones are equally covered by '+'s and '-'s. The threshold evaluation of ASP is dominated by '-'s. The reason for the non-optimum threshold is that the ASP histogram for a category with respect to different thresholds does not have a significant and clear peak which hints to the optimum threshold (e.g. histograms vary between 0.1 and 0.18). In contrast to ASP, the F1 and F2 histograms do have significant and clear peaks from which the optimum thresholds can clearly be extracted.

The scores in detail are listed here:

		F1		All Results		
Topic	#rel	#docs	score	min	med	max
1	51	2	6	-318	9	56
3	76	89	-33	-676	-6	28
4	80	109	27	-696	-12	27
5	7	1	3	-781	0	4
6	165	153	-171	-913	35	112
11	174	67	71	-156	52	82
12	292	234	102	-15	102	381
23	7	0	0	-621	0	0
24	42	1	-2	-69	-7	1
44	4	0	0	-1968	-2	3
54	174	73	119	-245	119	285
58	18	2	-4	-637	-4	0
62	401	43	24	-903	12	24
77	16	33	-66	-662	0	6
78	45	10	10	-583	3	30
82	82	86	-132	-132	32	78
94	193	5	5	-45	5	40
95	138	4	-3	-243	-3	32
100	197	77	66	-90	105	244

108	314	7	6	-40	37	140
111	566	349	252	-70	288	560
114	59	134	-78	-248	27	50
118	89	348	-611	-611	-165	17
119	85	3	-1	-294	-33	-1
123	62	29	-43	-1533	0	36
125	27	0	0	-1985	0	8
126	19	0	0	-150	3	14
128	333	0	0	-1780	0	49
142	229	963	-1361	-1361	-257	-10
148	260	25	70	-881	0	70
154	175	1	3	-1426	145	386
161	121	138	-26	-608	163	260
173	16	0	0	-1995	0	1
180	17	0	0	-1995	-4	0
185	18	11	-7	-1352	0	10
187	21	161	-307	-317	-165	0
189	890	548	209	0	180	837
192	7	0	0	-556	0	0
194	4	0	0	-1995	-2	0
202	627	40	120	-15	89	644
228	65	2	6	-460	3	18
240	131	9	-3	-1980	1	12
282	28	0	0	-68	0	9
10001	135	2	6	-8	6	26
10002	321	1	3	-383	6	40
10003	73	0	0	-52	0	15
10004	18	47	-89	-89	-2	9

		F2		All Results		
Topic	#rel	#docs	score	min	med	max
1	51	6	-32	-56	-20	57
3	76	388	-204	-269	-54	20
4	80	180	-5	-239	-43	26
5	7	31	-13	-82	-9	1
6	165	176	-211	-211	-47	120
11	174	317	59	-171	-5	62
12	292	468	135	-293	65	422

23	7	0	-7	-16	-8	-7
24	42	1	-43	-69	-48	-25
44	4	0	-4	-50	-5	5
54	174	76	20	-219	76	311
58	18	2	-20	-27	-20	-3
62	401	76	-282	-435	-349	-203
77	16	53	-69	-69	-16	-6
78	45	11	-21	-57	-21	21
82	82	89	-131	-137	-2	89
94	193	60	-118	-224	-185	-87
95	138	195	-238	-338	-139	-14
100	197	222	-34	-217	26	196
108	314	59	-198	-334	-184	240
111	566	1000	89	-576	84	621
114	59	117	9	-128	8	50
118	89	732	-701	-701	-209	-47
119	85	17	-92	-215	-102	-72
123	62	42	-79	-79	-49	24
125	27	0	-27	-56	-27	-8
126	19	0	-19	-37	-4	16
128	333	0	-333	-1113	-318	-220
142	229	1000	-664	-664	-327	-126
148	260	226	344	-286	-256	344
154	175	0	-175	-180	167	405
161	121	166	-7	-174	164	249
173	16	17	-23	-42	-20	-9
180	17	0	-17	-72	-19	-17
185	18	10	-18	-330	-18	-4
187	21	681	-642	-642	-176	-21
189	890	1000	-35	-886	-146	923
192	7	0	-7	-40	-14	-7
194	4	0	-4	-84	-7	-4
202	627	566	202	-627	-374	834
228	65	38	-58	-220	-43	-2
240	131	31	-112	-449	-118	-87
282	28	0	-28	-50	-27	-16
10001	135	0	-135	-158	-120	-76
10002	321	26	-247	-326	-286	-192
10003	73	0	-73	-648	-73	-18

10004	18	39	-52	-66	-19	-2

		ASP		All Results		
Topic	#rel	#docs	score	min	med	max
1	51	3	0.059	0.000	0.098	0.386
3	76	308	0.103	0.000	0.118	0.260
4	80	133	0.235	0.000	0.086	0.260
5	7	1	0.143	0.000	0.082	0.286
6	165	153	0.029	0.000	0.150	0.236
11	174	224	0.217	0.000	0.135	0.227
12	292	744	0.215	0.000	0.202	0.416
23	7	0	0.000	0.000	0.000	0.080
24	42	1	0.000	0.000	0.006	0.072
44	4	6	0.042	0.000	0.004	0.375
54	174	77	0.226	0.000	0.031	0.524
58	18	2	0.000	0.000	0.000	0.045
62	401	41	0.029	0.000	0.010	0.045
77	16	1000	0.001	0.000	0.000	0.125
78	45	1000	0.045	0.000	0.050	0.235
82	82	90	0.011	0.000	0.148	0.325
94	193	1000	0.032	0.000	0.021	0.072
95	138	5	0.001	0.000	0.017	0.153
100	197	245	0.133	0.000	0.197	0.398
108	314	408	0.157	0.000	0.148	0.246
111	566	759	0.198	0.000	0.233	0.394
114	59	106	0.219	0.000	0.187	0.305
118	89	543	0.007	0.000	0.056	0.097
119	85	3	0.004	0.000	0.031	0.053
123	62	44	0.009	0.000	0.022	0.215
125	27	0	0.000	0.000	0.000	0.083
126	19	15	0.004	0.000	0.022	0.237
128	333	9	0.003	0.000	0.006	0.056
142	229	1000	0.056	0.000	0.128	0.173
148	260	212	0.482	0.000	0.008	0.511
154	175	1	0.006	0.000	0.274	0.706
161	121	168	0.149	0.000	0.455	0.692
173	16	37	0.007	0.000	0.000	0.087
180	17	26	0.000	0.000	0.000	0.001

185	18	15	0.059	0.000	0.000	0.148
187	21	147	0.003	0.000	0.006	0.060
189	890	916	0.153	0.000	0.129	0.329
192	7	0	0.000	0.000	0.000	0.010
194	4	1	0.250	0.000	0.000	0.250
202	627	673	0.216	0.000	0.137	0.415
228	65	11	0.013	0.000	0.045	0.117
240	131	11	0.006	0.000	0.015	0.051
282	28	0	0.000	0.000	0.000	0.107
10001	135	2	0.015	0.000	0.015	0.074
10002	321	3	0.009	0.000	0.008	0.048
10003	73	0	0.000	0.000	0.000	0.068
10004	18	23	0.000	0.000	0.000	0.167

Acknowledgement Ulrich Kressel and Jürgen Franke, both members of our research group, helped a lot in computing the PCA for large matrices and with fruitful discussions.

References

[1] T. Bayer, U. Bohnacker, I. Renz, Information Extraction From Paper Documents, *Handbook of Optical Character Recognition and Document Image Analysis* (P.S.P. Wang and H. Bunke, Eds.), Singapore, 1997.

[2] T. Bayer, U. Kressel, H. Mogg-Schneider, and I. Renz, Categorizing Paper Documents - A Generic System for Domain and Language Independent Text Categorization, to appear in *Journal of Computer Vision and Image Understanding. Special Issue on Document Image Understanding and Retrieval*, 1998.

[3] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, 41(6), 1990.

[4] U. Kressel and J. Schürmann, Pattern Classification Techniques Based on Function Approximation, *Handbook of Optical Character Recognition and Document Image Analysis* (P.S.P. Wang and H. Bunke, Eds.), Singapore, 1997.

[5] J. Schürmann, *Pattern Classification*, Wiley, New York, 1996.

[6] H. Schütze, D. Hull, and J. Pederson, A Comparison of Classifiers and Document Representation for the Routing problem, *Proceedings, 18th Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, Seattle, WA, 1995.

[7] Y. Y. Yang and C. G. Chute, An Application of Least Squares Fit Mapping To Text Information Retrieval, *Proceedings, 16th Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, Pittsburgh, PA, 1993.

[8] Y. Y. Yang, Noise reduction in a statistical approach to text categorization, *Proceedings, 18th Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, Seattle, WA, 1995.

NATURAL LANGUAGE INFORMATION RETRIEVAL TREC-6 REPORT

TOMEK STRZALKOWSKI AND FANG LIN
GE Corporate Research & Development
Schenectady, NY 12301, USA

AND

JOSE PEREZ-CARBALLO
School of Communication, Information and Library Studies
Rutgers University
New Brunswick, NJ 04612

Abstract. Natural language processing techniques may hold a tremendous potential for overcoming the inadequacies of purely quantitative methods of text information retrieval, but the empirical evidence to support such predictions has thus far been inadequate, and appropriate scale evaluations have been slow to emerge. In this chapter, we report on the progress of the Natural Language Information Retrieval project, a joint effort of several sites led by GE Research, and its evaluation in the 6th Text Retrieval Conferences (TREC-6).

1. Introduction and Motivation

Recently, we noted a renewed interest in using NLP techniques in information retrieval, sparked in part by the sudden prominence, as well as the perceived limitations, of existing IR technology in rapidly emerging commercial applications, including on the Internet. This has also been reflected in what is being done at TREC: using phrasal terms and proper name annotations became a norm among TREC participants, and a special interest track on NLP took off for the first time in TREC-5.

In this paper we discuss particulars of the joint GE/Rutgers TREC-6 entry.

2. Stream-based Information Retrieval Model

The stream model was conceived to facilitate a thorough evaluation and optimization of various text content representation methods, including simple quantitative techniques as well as those requiring complex linguistic processing. Our system encompasses a number of statistical and natural language processing techniques that capture different aspects of document content: combining these into a coherent whole was in itself a major challenge. Therefore, we designed a distributed representation model in which alternative methods of document indexing (which we call "streams") are strung together to perform in parallel. Streams are built using a mixture of different indexing approaches, term extracting and weighting strategies, even different search engines.

The following term extraction steps correspond to some of the streams used in our system:

1. Elimination of stopwords: Original text words minus certain no-content and low-content stopwords are used to index documents. Included in the stopwords category are closed-class words such as determiners, prepositions, pronouns, etc., as well as certain very frequent words.
2. Morphological stemming: Words are normalized across morphological variants (e.g., "proliferation", "proliferate", "proliferating") using a lexicon-based stemmer. This is done by chopping off a suffix (*-ing, -s, -ment*) or by mapping onto root form in a lexicon (e.g., *proliferation* to *proliferate*).
3. Phrase extraction: Various shallow text processing techniques, such as part-of-speech tagging, phrase boundary detection, and word co-occurrence metrics are used to identify relatively stable groups of words, e.g., *joint venture.*
4. Phrase normalization: "Head+Modifier" pairs are identified in order to normalize across syntactic variants such as *weapon proliferation, proliferation of weapons, proliferate weapons*, etc., and reduce to a common "concept", e.g., **weapon+proliferate**.
5. Proper name extraction: Proper names are identified for indexing, including people names and titles, location names, organization names, etc.

The final results are produced by merging ranked lists of documents obtained from searching all streams with appropriately preprocessed queries, i.e., phrases for phrase stream, names for names stream, etc. The merging process weights contributions from each stream using a combination that was found the most effective in training runs. This allows for an easy combination of alternative retrieval and routing methods, creating a meta-search strategy which maximizes the contribution of each stream. Cornell's

SMART (Salton, 1989), Umass' Inquery (Croft et al., 19xx), and NIST's Prise (Harman & Candella, 1989) information retrieval systems were used as search engines for different streams.

Among the advantages of the stream architecture we may include the following:

- stream organization makes it easier to compare the contributions of different indexing features or representations. For example, it is easier to design experiments which allow us to decide if a certain representation adds information which is not contributed by other streams.
- it provides a convenient testbed to experiment with algorithms designed to merge the results obtained using different IR engines and/or techniques.
- it becomes easier to fine-tune the system in order to obtain optimum performance
- it allows us to use any combination of IR engines without having to adapt them in any way.

The notion of combining evidence from multiple sources is not new in information retrieval. Several researchers have noticed in the past that different systems may have similar performance but retrieve different documents, thus suggesting that they may complement one another. It has been reported that the use of different sources of evidence increases the performance of a hybrid system (see for example, (Callan et al., 1995); (Fox et al., 1993); (Saracevic and Kantor, 1988)). Nonetheless, the stream model used in our system is unique in that it explicitly addresses the issue of document representation as well as provides means for subsequent optimization.

3. Advanced Linguistic Streams

3.1. HEAD+MODIFIER PAIRS STREAM

Our linguistically most advanced stream is the head+modifier pairs stream. In this stream, documents are reduced to collections of word pairs derived via syntactic analysis of text followed by a normalization process intended to capture semantic uniformity across a variety of surface forms, e.g., "information retrieval", "retrieval of information", "retrieve more information", "information that is retrieved", etc. are all reduced to "retrieve+information" pair, where "retrieve" is a head or operator, and "information" is a modifier or argument. It has to be noted that while the head-modifier relation may suggest semantic dependence, what we obtain here is strictly syntactic, even though the semantic relation is what we are really after. This means in particular that the inferences of the kind where a *head+modifier* is taken as a specialized instance of *head*, are inherently risky,

because the head is not necessarily a semantic head, and the *modifier* is not necessarily a semantic modifier, and in fact the opposite may be the case. In the experiments that we describe here, we have generally refrained from semantic interpretation of head-modifier relationship, treating it primarily as an *ordered* relation between otherwise equal elements. Nonetheless, even this simplified relationship has already allowed us to cut through a variety of surface forms, and achieve what we thought was a non-trivial level of normalization. The apparent lack of success of linguistically-motivated indexing in information retrieval may suggest that we haven't still gone far enough.

In our system, the head+modifier pairs stream is derived through a sequence of processing steps that include:

1. Part-of-speech tagging
2. Lexicon-based word normalization (extended "stemming")
3. Syntactic analysis with TTP parser
4. Extraction of head+modifier pairs
5. Corpus-based disambiguation of long noun phrases

These steps are described briefly below. For details the reader is referred to past TREC articles, and other works, including (Strzalkowski, 1995) and (Strzalkowski et al., 1997).

3.1.1. *Part-of-speech tagging*

Part of speech tagging allows for resolution of lexical ambiguities in a running text, assuming a known general type of text (e.g., newspaper, technical documentation, medical diagnosis, etc.) and a context in which a word is used. This in turn leads to a more accurate lexical normalization or stemming. It also is a basis for a phrase boundary detection.

We used a version of Brill's rule based tagger (Brill, 1992) trained on Wall Street Journal texts to preprocess linguistic streams used by SMART. We also used BBN's stochastic POST tagger as part of our NYU-based Prise system. Both systems are based on the Penn Treebank Tagset developed at the University of Pennsylvania, and have compatible levels of performance.

3.1.2. *Lexicon-based word normalization*

Word stemming has been an effective way of improving document recall since it reduces words to their common morphological root, thus allowing more successful matches. On the other hand, stemming tends to decrease retrieval precision, if care is not taken to prevent situations where otherwise unrelated words are reduced to the same stem. In our system we replaced a traditional morphological stemmer with a conservative dictionary-assisted

suffix trimmer.[1]

The suffix trimmer performs essentially two tasks:

1. it reduces inflected word forms to their root forms as specified in the dictionary, and
2. it converts nominalized verb forms (e.g., "implementation", "storage") to the root forms of corresponding verbs (i.e., "implement", "store").

This is accomplished by removing a standard suffix, e.g., "stor+age", replacing it with a standard root ending ("+e"), and checking the newly created word against the dictionary, i.e., we check whether the new root ("store") is indeed a legal word.

3.1.3. *Syntactic analysis with TTP*

Parsing reveals finer syntactic relationships between words and phrases in a sentence, relationships that are hard to determine accurately without a comprehensive grammar. Some of these relationships do convey semantic dependencies, e.g., in *Poland is attacked by Germany* the subject+verb and verb+object relationships uniquely capture the semantic relationship of who attacked whom. The surface word-order alone cannot be relied on to determine which relationship holds. From the onset, we assumed that capturing semantic dependencies may be critical for accurate text indexing. One way to approach this is to exploit the syntactic structures produced by a fairly comprehensive parser.

TTP (Tagged Text Parser) is based on the Linguistic String Grammar developed by Sager (Sager, 1981) . The parser currently encompasses some 400 grammar productions, but it is by no means complete. The parser's output is a regularized parse tree representation of each sentence, that is, a representation that reflects the sentence's logical predicate-argument structure. For example, logical subject and logical object are identified in both passive and active sentences, and noun phrases are organized around their head elements. The parser is equipped with a powerful skip-and-fit recovery mechanism that allows it to operate effectively in the face of ill-formed input or under a severe time pressure. TTP has been shown to produce parse structures which are no worse than those generated by full-scale linguistic parsers when compared to hand-coded Treebank parse trees (Strzalkowski and Scheyen, 1996).

3.1.4. *Extracting head+modifier pairs*

Syntactic phrases extracted from TTP parse trees are head+modifier pairs. The head in such a pair is a central element of a phrase (main verb, main

[1]Dealing with prefixes is a more complicated matter, since they may have quite strong effect upon the meaning of the resulting term, e.g., "un- " usually introduces explicit negations.

noun, etc.), while the modifier is one of the adjunct arguments of the head. It should be noted that the parser's output is a predicate-argument structure centered around main elements of various phrases. The following types of pairs are considered: (1) a head noun and its left adjective or noun adjunct, (2) a head noun and the head of its right adjunct, (3) the main verb of a clause and the head of its object phrase, and (4) the head of the subject phrase and the main verb. These types of pairs account for most of the syntactic variants for relating two words (or simple phrases) into pairs carrying compatible semantic content. This also gives the pair-based representation sufficient flexibility to effectively capture content elements even in complex expressions. There are of course exceptions. For example, the threeword phrase "former Soviet president" would be broken into two pairs "former president" and "Soviet president", both of which denote things that are potentially quite different from what the original phrase refers to, and this fact may have potentially a negative effect on retrieval precision. This is one place where a longer phrase appears more appropriate. Below is a small sample of head+modifier pairs extracted (proper names are not included):

original text:
While serving in South Vietnam, a number of U.S. Soldiers were reported as having been exposed to the defoliant Agent Orange. The issue is veterans entitlement, or the awarding of monetary compensation and/or medical assistance for physical damages caused by Agent Orange.

head+modifier pairs:
damage+physical, cause+damage, award+assist, award+compensate, compensate+monetary, assist+medical, entitle+veteran

3.1.5. *Corpus-based disambiguation of long noun phrases*
The phrase decomposition procedure is performed after the first phrase extraction pass in which all unambiguous pairs (noun+noun and noun+adjective) and all ambiguous noun phrases are extracted. Any nominal string consisting of three or more words of which at least two are nouns is deemed structurally ambiguous. In the TREC corpus, about 80% of all ambiguous nominals were of length 3 (usually 2 nouns and an adjective), 19% were of length 4, and only 1% were of length 5 or more. The phrase decomposition algorithm has been described in detail in (Strzalkowski, 1995). The algorithm was shown to provide about 70% recall and 90% precision in extracting correct head+modifier pairs from 3 or more word noun groups in TREC collection texts. In terms of the total number of pairs extracted unambiguously from the parsed text, the disambiguation step recovers an

additional 10% to 15% of pairs, all of which would otherwise be either discarded or misrepresented.

3.2. SIMPLE NOUN PHRASE STREAM

In contrast to the elaborate process of generating the head+modifier pairs, unnormalized noun groups are collected from part-of-speech tagged text using a few regular expression patterns. No attempt is made to disambiguate, normalize, or get at the internal structure of these phrases, other than the stemming which has been applied to text prior to the phrase extraction step. The following phrase patterns have been used, with phrase length arbitrarily limited to the maximum 7 words:

1. a sequence of modifiers (adjectives, participles, etc.) followed by at least one noun, such as: "cryonic suspension", "air traffic control system";
2. proper noun sequences modifying a noun, such as: "u.s. citizen", "china trade";
3. proper noun sequences (possibly containing '&'): "warren commission", "national air traffic controller".

The motivation for having a phrase stream is similar to that for head+modifier pairs since both streams attempt to capture significant multi-word indexing terms. The main difference is the lack of normalization, which makes the comparison between these two streams particularly interesting.

3.3. NAME STREAM

In our system names are identified by the parser, and then represented as strings, e.g., south+africa. The name recognition procedure is extremely simple, in fact little more than the scanning of successive words labeled as proper names by the tagger ("np" and "nps" tags). Single-word names are processed just like ordinary words, except for the stemming which is not applied to them. We also made no effort to assign names to categories, e.g., people, companies, places, etc., a classification which is useful for certain types of queries (e.g., "To be relevant a document must identify a specific generic drug company"). In the TREC-5 database, compound names make up about 8% of all terms generated. A small sample of compound names extracted is listed below:

right+wing+christian+fundamentalism, gun+control+legislation,
u.s+government, exxon+valdez, plo+leader+arafat,
national+railroad+transportation+corporation,
suzuki+samurai+soft-top+4wd

TABLE 1. How different streams perform relative to one another (11-pt avg. Prec)

RUNS	*short queries*	*long queries*
Stems	0.1070	0.2684
Phrases	0.0846	0.2541
H+M Pairs	0.0405	0.1787
Names	0.0648	0.0753

3.4. STEMS STREAM

The stems stream is the simplest, yet the most effective of all streams, a backbone of the multistream model. It consists of stemmed single-word tokens (plus hyphenated phrases) taken directly from the document text (exclusive of stopwords). The stems stream provides the most comprehensive, though not very accurate, image of the text it represents, and therefore it is able to outperform other streams that we used thus far. We believe however, that this representation model has reached its limits, and that further improvement can only be achieved in combination with other text representation methods. This appears consistent with the results reported at TREC.

In addition, we use WordNet (Miller, 1980) to identify unambiguous single-sense words and give them premium weights as reliable discriminators. Many words, when considered out of context, display more than one sense in which they can be used. When such words are used in text they may assume any of their possible senses thus leading to undesired matches. This has been a problem for word based IR systems, and have spurred attempts at sense disambiguation in text indexing (Krovetz and Croft, 1992). Another way to address this problem is to focus on words that do not have multiple-sense ambiguities, and treat these as special, because they seem to be more reliable as content indicators. This modification has produced a slightly stronger stream.

The results in Table 1 are somewhat counter-intuitive, particularly the unexpectedly weak performance of H+M Pairs stream. While we have noticed that Phrases often outperform Pairs (cf. TREC-5 results), the difference was never this pronounced. One possible explanation is a worse than expected quality of parse structures generated by TTP, which may be related to sub-optimal setting of critical parameters, particularly the time-out value. We continue to investigate these results.

For streams using SMART indexing, we selected optimal term weighting

TABLE 2. Term weighting across streams using SMART

STREAM	*weighting scheme*
Stems	lnc.ntn
Phrases	ltn.ntn
H+M Pairs	ltn.nsn
Names	ltn.ntn

schemes from among a dozen or so variants implemented with version 11 of the system. These schemes vary in the way they calculate and normalize basic term weights. For example, in *lnc.ntn* scheme, *lnc* scoring (*log-tf, no-idf, cosine-normalization*) is applied to documents, and *ntn* scoring (*straight-tf, idf, nonormalization*) is applied to query terms. The selection of one scheme over another can have a dramatic effect on system's performance. For details the reader is referred to (Buckley, 1993).

4. Stream Merging and Weighting

The results obtained from different streams are lists of documents ranked in order of relevance: the higher the rank of a retrieved document, the more relevant it is presumed to be. In order to obtain the final retrieval result, ranking lists obtained from each stream have to be combined together by a process known as merging or fusion. The final ranking is derived by calculating the combined relevance scores for all retrieved documents. The following are the primary factors affecting this process:

1. document relevancy scores from each stream
2. retrieval precision distribution estimates within ranks from various streams, e.g., projected precision between ranks 10 and 20, etc.;
3. the overall effectiveness of each stream (e.g. measured as average precision on training data)
4. the number of streams that retrieve a particular document, and
5. the ranks of this document within each stream.

Generally, a stronger (i.e., better performing) stream will more effect on shaping the final ranking. A document which is retrieved at a high rank from such a stream is more likely to end up ranked high in the final result. In addition, the performance of each stream within a specific range of ranks is taken into account. For example, if phrases stream tends to pack relevant documents between the top 10th and 20th retrieved documents (but not so much into 1-10) we would give premium weights to the documents found

TABLE 3. Precision improvements over stems-only retrieval based on TREC-5 data

Streams merged	*short queries % change*	*long queries % change*
All streams	+5.4	+20.94
Stems+Phrases+Pairs	+6.6	+22.85
Stems+Phrases	+7.0	+24.94
Stems+Pairs	+2.2	+15.27
Stems+Names	+0.6	+2.59

in this region of phrase-based ranking, etc. Table 3 gives some additional data on the effectiveness of stream merging. Further details are available in our TREC-5 conference article (Strzalkowski et al., 1997).

Note that long text queries benefit more from linguistic processing.

4.1. INTER-STREAM MERGING USING PRECISION DISTRIBUTION ESTIMATES

We used the following two principal sources of information about each stream to weigh their relative contributions to the final ranking:

- an actual ranking obtained from a training run (training data, old queries);
- an estimated retrieval precision at certain ranges of ranks.

Precision estimates are used to order results obtained from the streams, and this ordering may vary at different rank ranges. Table 4 shows precision estimates for selected streams at certain rank ranges as obtained from a training collection derived from TREC-4 data.

The final score of a document (d) is calculated using the following formula:

$$finalscore(d) = \sum_{i=1...N} A(i) \times score(i)(d) \times prec(\{ranks(i)|rank(i,d) \in ranks(i)\})$$

where N is the number of streams; $A(i)$ is the stream coefficient; and $score(i)(d)$ is the normalized score of the document against the query within the stream i; $prec(ranks(i))$ is the precision estimate from the precision distribution table for stream i; and $rank(i,d)$ is the rank of document d in stream i.

TABLE 4. Precision distribution estimates for selected streams

RANKS	STEMS	PHRASES	PAIRS	NAMES
1-5	0.49	0.45	0.33	0.23
6-10	0.42	0.38	0.27	0.18
11-20	0.37	0.32	0.23	0.13
21-30	0.33	0.28	0.21	0.10
31-50	0.27	0.25	0.17	0.08
51-100	0.19	0.17	0.12	0.06
101-200	0.12	0.11	0.08	0.04

TABLE 5. Stream merging coefficient structures used in TREC-5

	STREAMS			
RUNS	stems	phrases	pairs	names
ad-hoc gerua1	4	3	3	1
ad-hoc gerua3	5	3	3	1
routing gerou1	4	3	3	1
routing gesri2	4	3	3	1

4.2. STREAM COEFFICIENTS

For merging purposes, streams are assigned numerical coefficients, referred to as A(i) above, that have two roles:

1. Control the relative contribution of a document score assigned to it within a stream when calculating the final score for this document. This applies primarily to streams producing normalized document scores, such as SMART.
2. Change stream-to-stream document score relationships for un-normalized ranking system, e.g., PRISE.

An example of a coefficient structure is shown below. They are obtained empirically to maximize the performance of any specific combination of streams. Table 5 summarizes stream coefficient structures used in TREC-5 experiments. Typically, a new combination was created for a given collection, a retrieval mode (ad-hoc vs. routing) and the search engines used.

5. Query Expansion Experiments

5.1. WHY QUERY EXPANSION?

The purpose of query expansion is to make the user query resemble more closely the documents it is expected to retrieve. This includes both content, as well as some other aspects such as composition, style, language type, etc. If the query is indeed made to resemble a "typical" relevant document, then suddenly everything about this query becomes a valid search criterion: words, collocations, phrases, various relationships, etc. Unfortunately, an average search query does not look anything like this, most of the time. It is more likely to be a statement specifying the semantic criteria of relevance. This means that except for the semantic or conceptual resemblance (which we cannot model very well as yet) much of the appearance of the query (which we can model reasonably well) may be, and often is, quite misleading for search purposes. Where can we get the right queries?

In today's information retrieval, query expansion usually pertains content and typically is limited to adding, deleting or re-weighting of terms. For example, content terms from documents judged relevant are added to the query while weights of all terms are adjusted in order to reflect the relevance information. Thus, terms occurring predominantly in relevant documents will have their weights increased, while those occurring mostly in non-relevant documents will have their weights decreased. This process can be performed automatically using a relevance feedback method, e.g., (Rocchio, 1971), with the relevance information either supplied manually by the user (Harman, 1988), or otherwise guessed, e.g. by assuming top 10 documents relevant, etc. (Buckley, et al., 1995). A serious problem with this content-term expansion is its limited ability to capture and represent many important aspects of what makes some documents relevant to the query, including particular term co-occurrence patterns, and other hard-to-measure text features, such as discourse structure or stylistics. Additionally, relevance-feedback expansion depends on the inherently partial relevance information, which is normally unavailable, or unreliable. Other types of query expansions, including general purpose thesauri or lexical databases (e.g., Wordnet) have been found generally unsuccessful in information retrieval (cf. (Voorhees, 1993); (Voorhees, 1994)).

An alternative to term-only expansion is a full-text expansion which we tried for the first time in TREC-5. In our approach, queries are expanded by pasting in entire sentences, paragraphs, and other sequences directly from any text document. To make this process efficient, we first perform a search with the original, un-expanded queries (short queries), and then use top N (10, 20) returned documents for query expansion. These documents are not judged for relevancy, nor assumed relevant; instead, they are scanned for

passages that contain concepts referred to in the query. Expansion material can be found in both relevant and non-relevant documents, benefitting the final query all the same. In fact, the presence of such text in otherwise non-relevant documents underscores the inherent limitations of distribution-based term reweighting used in relevance feedback. Subject to some further "fitness criteria", these expansion passages are then imported verbatim into the query. The resulting expanded queries undergo the usual text processing steps, before the search is run again.

Full-text expansion can be accomplished manually, as we did initially to test feasibility of this approach in TREC-5, or semi-automatically, as we tried this year with excellent results. Our goal is to fully automate this process. (We did try an automatic expansion in TREC-5, but it was very simplistic and not very sucessful, cf. our TREC-5 report.)

The initial evaluations indicate that queries expanded manually following the prescribed guidelines are improving the system's performance (precision and recall) by as much as 40% or more. This appear to be true not only for our own system, but also for other systems: we asked other groups participating in TREC-5 to run search using our expanded queries, and they reported nearly identical improvements. Below, we describe the three different query expansion techniques explored in TREC-6.

5.2. SUMMARIZATION-BASED QUERY EXPANSION

We used an automatic text summarizer to derive query-specific summaries of documents returned from the first round of retrieval. The summaries were usually 1 or 2 consecutive paragraphs selected from the original document text. The purpose was to demonstrate, in a quick-read abstract, why a given document has been retrieved. If the summary appeared relevant and moreover captured some new aspect of relevant information, then it was pasted into the query. Note that it wasn't important if the document itself was relevant.

The summaries were produced automatically using GE Summarizer-Tool, a prototype developed for Tipster Phase 3 project. It works by extracting passages from the document text, and producing perfectly readable, very brief summaries, at about 5 to 10% of original text length.

A preliminary examination of TREC-6 results indicate that this mode of expansion is at least as effective as the purely manual expansion used in TREC-5. This is a very good news, since we now appear to be a step closer to an automatic expansion. The human-decision factor has been reduced to an accept/reject decision for expanding the search query with a summary – no need to read the whole document in order to select expansion passages.

5.3. EXTRACTION-BASED QUERY EXPANSION

We used automatic information extraction techniques to score text passages for presence of concepts (rather than keywords) identified by the query. Small extraction grammars were manually constructed for 23 out of 47 routing queries. Using SRI's FASTUS information extraction system, we selected highest score sentences from known relevant documents in the training corpus. Please note that this was a routing run, and the setup was somewhat different than in other query expansion runs. In particular, there was only one run against the test collection (the routing mode allows no feedback).

This run was constructed in collaboration with SRI's team. SRI has developed FASTUS grammars, run FASTUS over the training documents, scored each sentence, and sent the sentences to GE. GE team applied stream model processing to the queries, run the queries against the test collection, and submitted the results to NIST.

5.4. INTERACTIVE QUERY EXPANSION WITH INQUERY

The results produced at Rutgers were obtained using an interactive system. We believe that through interaction with the system and the database the user can create significantly better queries. The support to the user provided by the interface in order to build better queries is at least as important as any other part of the system.

In our previous contributions we have devoted very significant resources in terms of processing power and time to the creation of better document representations. In particular, we have applied NLP techniques to thousands of megabytes of text in order to add less ambiguous terms to the document representation. In the interaction experiment we attempted to move processing power and "intelligence" from the representation to the interface. What we are trying to do is to spend a few tenths of a second executing even more sophisticated techniques (including, in future interfaces, NLP) on the query instead of days processing several gugabytes of the corpus in order to generate a better representation.

A new user interface for InQuery, called RUINQ2, was developed at Rutgers for this experiment. This is a variation of RUINQ, the InQuery interface developed for use in the interactive track experiments reported by the Rutgers team (see Rutgers paper in these proceedings).

RUINQ2 supports the use of negative and positive feedback. The user is shown a list of 10 document titles at a time. The user can scroll to see another 10 as many times as needed. Any number of the titles presented can be declared either relevant or non relevant by the user (by clicking next to the title). When a document is declared relevant (non relevant) some terms

are offered to the user on a positive (negative) feedback window. The user can add to the query any number of terms from those windows by clicking on the desired term.

RUINQ2 also supported the use of phrases (any sequence of words entered by the user inside double quotes) and required terms (preceeded by a plus sign).

The interactive run was created in order to have a baseline to compare query expansion using automatically generated summaries, with query expansion using interaction with document text plus negative/positive feedback. Further experiments based on more refined user interfaces for both systems should help us answer questions such as: which system is easier to use, which one allows users to create queries faster and which system helps user create more effective queries.

The interactive run was created by allowing a single user (one of the authors) who had never seen the topics before, to interact with the system for no more than 15mins per topic in order to build the corresponding query. When the user was satisfied with the query he would click on a button that would print out the rankings of (at most) 1000 documents in TREC format. In several cases less than 1000 documents were found.

We discovered a bug in the program after we had submmited the results. The ranking printed out began by the first document displayed on the document title screen at the moment the user decided to print out the rankings (as opposed to the first document of the ranking). So, if the user was looking at the second page of document titles at the time he printed out the ranking, the first 10 documents of the ranking were not printed. This happened with about 8 queries. The corrected results will be presented in our talk at the conference.

6. SUMMARY OF RESULTS

6.1. AD-HOC RUNS

Ad-hoc retrieval is when an arbitrary query is issued to search a database for relevant documents. In a typical ad-hoc search situation, a query is used once, then discarded, thus leaving little room for optimization. Our ad-hoc experiments were conducted in several subcategories, including automatic, manual, and using different sizes of databases and different types of queries. An automatic run means that there was no human intervention in the process at any time. A manual run means that some human processing was done to the queries, and possibly multiple test runs were made to improve the queries. A short query is derived using only one section of a TREC-5 topic, namely the DESCRIPTION field. A full query is derived from any or all fields in the topic. An example TREC-5 query is show below; note that

the Description field is what one may reasonably expect to be an initial search query, while Narrative provides some further explanation of what relevant material may look like. The Topic field provides a single concept of interest to the searcher; it was not permitted in the short queries.

< *top* >
< *num* > Number: 324
< *title* > Argentine/British Relations
< *desc* > Description:
Define Argentine and British international relations
< *narr* > Narrative:
It has been 15 years since the war between Argentina and the United Kingdom in 1982 over sovereignty in the Falkland Islands. A relevant report will describe their relations after that period. Any kind of international contact between the two countries is relevant, to include commercial, economic, cultural, diplomatic, or military exchanges. Negative reports on the absence of such exchanges are also desirable. Reports containing information on direct exchanges between Argentina and the Falkland Islands are also relevant.
< */top* >

Table 6 summarizes selected runs performed with our NLIR system on TREC-6 database using 50 queries numbered 301 through 350. The SMART baselines were produced by Cornell-SaBir team using version 11 of the system. The rightmost column is an unofficial rerun of the GERUA1 after fixing of a simple bug. Table 7 compares the performance of UMass' InQuery system on the same set of queries, and the same database. Note the consistently large improvements in retrieval precision attributed to the expanded queries.

6.2. ROUTING RUNS

Routing is a process in which a stream of previously unseen documents are filtered and distributed among a number of standing profiles, also known as routing queries. In routing, documents can be assigned to multiple profiles. In categorization, a type of routing, a single best matching profile is selected for each document. Routing is harder to evaluate in a standardized setup than the retroactive retrieval because of its dynamic nature, therefore a simulated routing mode has been used in TREC. A simulated routing mode (TREC-style) means that all routing documents are available at once, but the routing queries (i.e., terms and their weights) are derived with respect to a different training database, specifically TREC collections from previous evaluations. This way, no statistical or other collection-specific information about the routing documents is used in building the profiles, and the participating systems are forced to make assumptions about the routing documents just like they would in real routing. However, no real routing occurs, and the prepared routing queries are run against the rout-

TABLE 6. Precision improvement in NLIR system vs. SMART (v.11) baselines

queries:	full	full	man long	man long	man long-1
PREC.	SMART	Best NL	SMART	Best NL	Best NL
11pt. avg	0.1429	0.1837	0.2672	0.2783	0.2859
%change		+28.5	+87.0	+94.7	+100.0
@10 docs	0.3000	0.3840	0.5060	0.5200	0.5200
%change		+28.0	+68.6	+73.3	+73.3
@30 docs	0.2387	0.2747	0.3887	0.3933	0.3940
%change		+15.0	+62.8	+64.7	+65.0
@100 doc	0.1600	0.1736	0.2480	0.2598	0.2574
%change		+8.5	+55.0	+62.3	+60.8
Recall	0.57	0.53	0.61	0.58	0.62
%change		-7.0	+7.0	+1.7	+8.7

TABLE 7. Results for UMass' InQuery (no NL indexing)

PREC.	automatic full queries (T+D)	manual long queries
11pt.avg	0.2103	0.3057
%change		+45.0
@20 docs	0.3620	0.4510
%change		+25.0
R-Prec	0.2461	0.3327
%change		+35.0

ing database much the same way they would be in an ad-hoc retrieval. Documents retrieved by each routing query, ranked in order of relevance, become the content of its routing bin.

6.2.1. *Query development against the training collection*
In Smart routing, automatic relevance feedback was performed to build routing queries using the training data available from previous TRECs. The routing queries, split into streams, were then run against stream-indexed routing collection. The weighting scheme was selected in such a way that no collection-specific information about the current routing data has been used. Instead, collection-wide statistics, such as *idf* weights, were those

TABLE 8. Precision averages for 47 routing queries

STREAMS	11pt. Prec	At 5 docs	At 10 docs	R-Prec
main routing, gerou1	0.2702	0.5532	0.4787	0.3176
query expansion gesri2	0.2458	0.5447	0.4894	0.2906
reranked gerou1, srige1	0.2730	0.5574	0.5021	0.3126

derived from the training data. The routing was carried out in the following four steps:

1. A subset of the previous TREC collections was chosen as the training set, and four index streams were built. Queries were also processed and run against the indexes. For each query, 1000 documents are retrieved. The weighting schemes used were: lnc.ltc for stems, ltc.ntc for phrases, ltc.ntc for head+modifier pairs, and ltc.ntc for names.
2. The final query vector was then updated through an automatic feedback step using the known relevance judgements. Up to 350 terms occurring in the most relevant documents were added to each query. Two alternative expanded vectors were generated for each query using different sets of Roccio parameters.
3. For each query, the best performing expansion was retained. These were submitted to NIST as official routing queries.
4. The final queries were run against the four-stream routing test collection and retrieved results were merged.

6.2.2. *Query expansion via sentence extraction*
This run was described in the preceding section on query expansion.

6.2.3. *Re-ranking using rescoring via extraction*
This run was created at SRI using the output from GE's main routing run. SRI's FASTUS (Hobbs et al., 1996) was used to score documents retrieved and rerank them if they contained concepts asked for in the query. For details of this run please refer to SRI's chapter (Bear & Israel, this volume).

The results of using information extraction techniques are shown in Table 8 and compared to our main routing run. We note a slight improvement in average precision, and a more definite precision improvement near the top of the ranking in FASTUS rescoring run. This is only a first attempt at a serious-scale experiment of this kind, and the results are definitely encouraging.

7. CONCLUSIONS

We presented in some detail our natural language information retrieval system consisting of an advanced NLP module and a 'pure' statistical core engine. While many problems remain to be resolved, including the question of adequacy of term-based representation of document content, we attempted to demonstrate that the architecture described here is nonetheless viable. In particular, we demonstrated that natural language processing can now be done on a fairly large scale and that its speed and robustness has improved to the point where it can be applied to real IR problems.

The main observation to make is that thus far natural language processing has not proven as effective as we would have hoped in to obtain better indexing and better term representations of queries. Using linguistic terms, such as phrases, head-modifier pairs, names, does help to improve retrieval precision, but the gains remain quite modest. On the other hand, full text query expansion works remarkably well, and even more so in combination with linguistic indexing. Our main effort in the immediate future will be to explore ways to achieve at least partial automation of this process. Using information extraction techniques to improve retrival either by building better queries, or by reorganizing the results is another promising line of investigation.

Acknowledgements We would like to thank Donna Harman for making the NIST's PRISE system available to this project since the beginning of TREC. We also thank Chris Buckley for helping us to understand the inner workings of SMART. We would like to thank Ralph Weischedel for providing and assisting in the use of the BBN's part of speech tagger. Finally, thanks to SRI's Jerry Hobbs, David Israel and John Bear for their collaboration on joint experiments. This paper is based upon work supported in part by the Defense Advanced Research Projects Agency under Tipster Phase-3 Contract 97-F157200-000.

References

Brill, Eric. 1992. "A Simple Rule-Based Part Of Speech Tagger." Proceedings of the Third Conference on Applied Computational Linguistics (ANLP).

Buckley, Chris, Amit Singhal, Mandar Mitra, Gerard Salton. 1995. "New Retrieval Approches Using SMART: TREC 4". Proceedings of the Fourth Text REtrieval Conference (TREC-4), NIST Special Publication 500-236.

Buckley, Chris. 1993. "The Importance of Proper Weighting Methods." Human Language Technology, Proceedings of the workshop, Princeton, NJ. Morgan-Kaufmann, pp. 349-352.

Callan, Jamie, Zhihong Lu, and W. Bruce Croft. 1995. "Searching Distributed Collections with Inference Networks." Proceedings of ACM SIGIR'95. pp. 21-29.

Fox, Ed, M. Koushik, J. Shaw, R. Modlin, and D. Rao. 1993. "Comining Evidence from Multiple Searches." Proceedings of First Text Retrieval Conference (TREC-1), NIST Special Publication 500-207, National Institute of Standards and Technology, Gaithersburg, MD. pp. 319-328.

Harman, Donna. 1988. "Towards interactive query expansion." Proceedings of ACM SIGIR-88, pp. 321-331.

Hobbs, Jerry, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Andrew Kehler, Mark Stickel, and Mabry Tyson. 1996. "SRI's Tipster II Project." Advances in Text Processing, Tipster Progran Phase 2. Morgan Kaufmann, pp. 201-208.

Krovetz, Robert and W. Bruce Croft. 1992. "Lexical ambiguity and information retrieval." *ACM Transactions on Information Systems*, 10(2), pp. 115-141.

Rocchio, J. J. 1971. "Relevance Feedback in Informatio Retrieval." In Salton, G. (Ed.), *The SMART Retrieval System*, pp. 313-323. Prentice Hall, Inc., Englewood Cliffs, NJ.

Sager, Naomi. 1981. *Natural Language Information Processing.* Addison-Wesley.

Saracevic, T., Kantor, P. 1988. "A Study of Information Seeking and Retrieving. III. Searchers, Searches, and Overlap." *Journal of the American Society for information Science*, 39(3):197-216.

Strzalkowski, Tomek, Louise Guthrie, Jussi Karlgren, Jim Leistensnider, Fang Lin, Jose Perez-Carballo, Troy Straszheim, Jin Wang, and Jon Wilding. 1997. "Natural Language Information Retrieval: TREC-5 Report." Proceedings of TREC-5 conference.

Strzalkowski, Tomek, and Peter Scheyen. 1993. "An Evaluation of TTP Parser: a preliminary report." Proceedings of International Workshop on Parsing Technologies (IWPT-93), Tilburg, Netherlands and Durbuy, Belgium, August 10-13.

Strzalkowski, Tomek and Jose Perez Carballo. 1994. "Recent Developments in Natural Language Text Retrieval." Proceedings of the First Text REtrieval Conference (TREC-2), NIST Special Publication 500-215, National Institute of Standards and Technology, Gaithersburg, MD. pp. 123-136.

Strzalkowski, Tomek. 1995. "Natural Language Information Retrieval" *Information Processing and Management*, Vol. 31, No. 3, pp. 397-417. Pergamon/Elsevier.

Strzalkowski, Tomek, and Peter Scheyen. 1996. "An Evaluation of TTP Parser: a preliminary report." In H. Bunt, M. Tomita (eds), *Recent Advances in Parsing Technology*, Kluwer Academic Publishers, pp. 201-220.

Voorhees, Ellen M. 1994. "Query Expansion Using Lexical-Semantic Relations." Proceedings of ACM SIGIR'94, pp. 61-70.

Voorhees, Ellen M. 1993. "Using WordNet to Disambiguate Word Senses for Text Retrieval." Proceedings of ACM SIGIR'93, pp. 171-180.

Using Information Extraction to Improve Document Retrieval

John Bear, David Israel, Jeff Petit, and David Martin
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

January 9, 1998

1 Abstract

We describe an approach to applying a particular kind of Natural Language Processing (NLP) system to the TREC routing task in Information Retrieval (IR). Rather than attempting to use NLP techniques in indexing documents in a corpus, we adapted an information extraction (IE) system to act as a post-filter on the output of an IR system. The IE system was configured to score each of the top 2000 documents as determined by an IR system and on the basis of that score to rerank those 2000 documents. One aim was to improve precision on routing tasks. Another was to make it easier to write IE grammars for multiple topics.

2 Introduction

Researchers have pursued a variety of approaches to integrating natural language processing with document retrieval systems. The central idea in the literature is that some, perhaps shallow variant of the kind of syntactic and semantic analysis performed by general-purpose natural language processing systems can provide information useful for improving the indexing, and thus the retrieval, of documents. [SparckJones1992, Lewis1992, Hearst1992] The work in this area has seen some success, but significant performance improvements have yet to be demonstrated. [Faloutsos and Oard1996] We have pursued a different hypothesis, that an information extraction (IE) system can be pipelined with a document retrieval system in such a way as to improve performance on routing tasks.

The goal of a document retrieval system, as embodied in the routing task of TRECs [Harman1996], is to consult a large database of documents and return a subset of documents ordered by decreasing likelihood of being relevant to a particular topic. In the TREC6 routing task, a document retrieval system returns the 1000 documents it judges most likely to be relevant to a query out

of a database of roughly one million documents. A system performs well if a high proportion of the articles returned, high relative to the ratio of relevant articles in the corpus, are relevant to the topic, and if the relevant articles are ranked earlier in its ordering than the irrelevant ones.

The goal of an IE system, as embodied in the scenario template task of MUCs [Grishman and Sundheim1995], is to consult a corpus of documents, usually smaller than those involved in document retrieval tasks, and extract prespecified items of information. (In MUC-6, for instance, the test corpus consisted of 100 newspaper articles.) Such a task might be defined, for instance, by specifying a template schema instances of which are to be filled automatically on the basis of a linguistic analysis of the texts in the corpus.

A system performs well to the extent that the material it extracts captures the relevant information in the documents. Note that if one were to apply the distinction between ad hoc and routing queries to the MUC scenario template task, it would be classified as a routing query; the task is known in advance and it is assumed that IE systems will have been especially tuned to the task.

Our approach to using NLP techniques for IR was to adapt an IE system, SRI's FASTUS system [Appelt et al.1995], to enable us to write small grammars for many topics and to use those grammars as queries to be run against the top 2000 documents for those topics, as determined by an IR system—in our case GE's version of SMART. In the end we were able to produce grammars for 23 topics.

As noted above, the output of an IR system for a given topic on the routing task is a list of the documents ordered by decreasing likelihood of relevance. Our adaptation of FASTUS involved having each grammar rule that matched some segment of an article assign a score to that segment. We then summed the scores to get a total for the article.

For each of the 23 topics for which we had written grammars, we had FASTUS process each article in GE's 2000 top articles for that topic and rank them by score. The highest-scoring articles were ranked first, and importantly, in the case of ties we used GE's order. For the other 24 topics, we submitted GE's top 1000 articles.

In the following sections, we will describe, first, the FASTUS Information Extraction System and then the main features of the adaptation of FASTUS to the current IR effort. We end with a brief summary and some tentative conclusions.

3 Background

3.1 FASTUS

SRI's FASTUS system is based on a cascade of finite-state transducers that compute the transformation of text from sequences of characters to domain

templates. Each transducer (or "phase") in FASTUS takes the output of the previous phase and maps it into structures that constitute the input to the next phase, or in the case of the final phase, that contain the domain template information that is the output of the extraction process. A typical FASTUS application might employ the following sequence of phases, although the number of transducers in different applications may vary.

1. *Tokenizer.* This phase accepts a stream of characters as input, and transforms it into a sequence of tokens.

2. *Multiword Analyzer.* This phase is generated automatically by the lexicon to recognize token sequences (like "because of") that are combined to form single lexical items.

3. *Name Recognizer.* This phase recognizes word sequences that can be unambiguously identified as names from their internal structure (like "ABC Corp." and "John Smith").

4. *Parser.* This phase constructs basic syntactic constituents of the language, consisting only of those that can be nearly unambiguously constructed from the input using finite-state rules (i.e., noun groups, verb groups, and particles).

5. *Combiner.* This phase produces larger constituents from the output of the parser when it can be done fairly reliably on the basis of local information. Examples are possessives, appositives, "of" prepositional phrases ("John Smith, 56, president of IBM's subsidiary"), coordination of same-type entities, and locative and temporal prepositional phrases.

6. *Domain or Clause-Level Phase.* The final phase recognizes the particular combinations of subjects, verbs, objects, prepositional phrases, and adjuncts that are necessary for correctly filling the templates for a given IE task.

The rules for each phase are specified in SRI's pattern language, called FASTSPEC. The rules take the form of regular productions that are translated automatically into finite-state machines by an optimizing compiler.

3.2 Adapting FASTUS for IR

The design of FASTUS was motivated by the design of MUC style scenario template tasks: a fairly narrowly defined prespecified information requirement was posed and up to a month's effort was devoted to writing application grammars to answer that requirement. In writing the grammars for MUCs 4 and 5 very little thought was given to the various ways in which greater generality of application might be built into the system. This changed with MUC6; we began work aimed toward making it easier to apply FASTUS to new topics.

For MUC6 we developed an approach that involved writing general, application-independent, clause-level patterns for which we would then write application-specific instances; typically, these instances were tied to the argument structure of the topic-relevant verbs. (For MUC6, where the task involved recognizing high-level management changes, these verbs included "resign", "succeed", "replace".) Given that we already had good reasons for extending this separation between application-independent rules and application-specific instances to earlier phases of FASTUS, in particular to the Parser and Combiner, the TREC routing task represented an extremely useful testbed for these adaptations.

Consider topic #12, for example. We want to recognize the various ways in which the simple predication *pollute(x, body-of-water)* might be expressed and then to automatically generate patterns to parse:

- full clauses in the Domain phase
  - "they polluted the stream"
  - "the reservoir has been contaminated"
- complex noun phrases in the Combiner phase
  - "the contamination of the creek"
  - "the bay's pollution"
- compound nouns in the Parser phase
  - "the water pollution"
  - "the polluted lake"

We give as an example the general pattern for the first of the two complex noun phrases. In such phrases, the object of the "of" phrase is the object of the event expressed by the head of the noun phrase ("contamination"):

```
ComplexNP  -->  ({NP[??subj] | NP[??obj]}   P[subcat=gen])
                {V-ING[TRANS,??head] | NP[TRANS,??head]}
                { P["of"]   NP[??obj] |
                  P["by"]  NP[??subj] |
                  P[??prep1] NP[??pobj1] |
                  P[??prep2] NP[??pobj2] }*   ;
  ??semantics  ;;
```

The topic-specific instance is as follows:

```
Instantiate
OfNP
??label = combiner-1-pollute
??subj = chemical
??head = pollute
??obj = body-of-water
??semantics =  weight = (assign-weight ((subj && obj) 10000)| (obj 1000)) ;;
```

The topic-specific instance can be thought of as a collection of macro definitions. During grammar-compilation, the "macro calls" in the patterns are expanded. In the example above, the string "??subj" is replaced by "chemical"; "??head", by "pollute", and so on. The resulting instantiated pattern is shown below:

```
ComplexNP  -->  ({NP[chemical] | NP[body-of-water]}   P[subcat=gen])
                {V-ING[TRANS, pollute] | NP[TRANS, pollute]}
                { P["of"]   NP[body-of-water] |
                  P["by"]  NP[chemical] }* ;
weight = (assign-weight ((subj && obj) 10000)| (obj 1000))
                ;;
```

Items in square brackets represent constraints on the phrase. For instance, "stream", "river" and "reservoir" are all nouns with the lexical feature *body-of-water* and only noun phrases with such nouns as heads satisfy the constraints on NP's in the rule instance.

3.3 An Initial Experiment

Having extended the method of general rules and application-specific instances to the Parser and Combiner, we were in a position to write grammars for multiple topics. We modeled our approach on an experiment we had performed running output from the INQUERY IR system through the MUC6 version of FASTUS.

The MUC-6 scenario template task is quite similar to TREC topic #15: "Document will announce the appointment of a new CEO and/or the resignation of a CEO of a company." In essence, the only difference between the MUC6 task and TREC topic #15 is that the latter is limited to the position of CEO. INQUERY was run with TREC topic #15 as an ad hoc query, producing a set of 1000 text documents it deemed most likely to be relevant, and ranking them in order from most likely relevant to least likely. Both the document set and the ordering served as inputs to FASTUS.

We tried two different schemes for using the information from FASTUS to reorder the input list. Both involved configuring the MUC6 grammar to assign scores to phrases based on correlation of phrase type with relevance. In

one scheme, we assigned scores to patterns manually, based on intuitions as to differential contributions to relevance judgments; in the second, a probabilistic model for the relevance of a document was inferred from a set of training data

As a basis for the first experiment, we picked 100 articles from the middle of the ordered set that INQUERY produced (in particular, articles ranked 401 through 500). The templates that the FASTUS MUC-6 system produced from those articles were examined to identify criteria for assigning a relevance rank to an article. We then had FASTUS assign a numerical score from 0.1 to 1000 to the templates that it produced for a phrase as follows:

1. CEO + person name + company name $\mapsto$ 1000
2. CEO + company name $\mapsto$ 100
3. CEO + person name $\mapsto$ 10
4. CEO + transition verb $\mapsto$ 1
5. CEO + BE verb $\mapsto$ 0.1

The score of a phrase was taken to be the sum of the scores of the templates created from that phrase; the scores from the phrases were summed to yield an article's score.

For the second experiment, we asked how a system for automatically identifying features concerning the output of FASTUS and determining the relative strengths of these features, would compare with the results obtained by the manually tuned system. A probabilistic model for the relevance of a document was inferred from a set of training data.

The results of these initial experiments [Kehler1996] were encouraging enough to motivate us to try both a larger and a more realistic experiment: one involving routing queries for many topics, none of which could have as much effort put into developing queries/grammars for it as was involved in producing the MUC6 application.

4 TREC6

For TREC6 we teamed with GE. They provided us with a ranked list of 2,000 documents for each query (using their version of SMART). We developed grammars for 23 of the 47 topics. For these 23 topics, FASTUS ran over the 2,000 articles, reordered them, and truncated to 1000. For the other 24 topics, we simply truncated GE's ordering at 1000 documents.

As in our first experiments, the reordering is achieved by having patterns—that is, instances (see example above)—assign a score to the segment (phrase) of an article successfully matched against. An article's total score is the sum of the scores of all the patterns that matched against phrases in that article.

As before, we broke ties by maintaining GE's relative order within a class of articles with identical scores.

For each topic we read a small number of relevant articles (10-15), constructed a topic-specific grammar by writing instances of the kind exemplified above, and then ran the grammar over some portion of the training data. Whenever a pattern matched a phrase, the phrase was recorded as being either a correct match or a false positive. We would review both sets of phrases, look at some of the relevant articles that were missed, and revise the grammar. After a small number of iterations of this kind, we would declare the grammar done, and move on to the next topic.

For training, we used a subset of the TREC6 training data, but we did not run over any of the results of GE's output on that corpus. We return to this point in our concluding section.

The scores were assigned with a threshold score in mind. An article had to contain at least one pattern that had a score of 1000 or above to be moved toward the beginning of the ordering; scores below 1000 had no effect on the order and were used solely for diagnostics. There was one exception to this general rule. For topic #11 (the space program), we tried the following mini-experiment: phrases were assigned maximum scores of 250, so that at least four matching phrases were needed to move an article to the front of the ranking. This was intended to handle cases where we could find no especially reliable phrasal indicators of relevance. Another way to put this is that this method is a crude approximation to a statistical approach based on co-occurrence data.

When writing the grammars our approach was to aim for high precision and to sacrifice recall when we were in a position to make a precision/recall tradeoff.

5 Results

As noted above, we were able to write grammars for 23 of the topics. Most of these grammars were written by a Stanford undergraduate who was, at the outset, completely unfamiliar with FASTUS. He spent about 5 to 6 hours per topic.

Overall we improved the average precision very slightly over our input, from 27% to 27.3%. We have included a graph, Figure 1, of precision versus recall for the 23 topics. The overall results of the combined system for average precision are: 12 topics above the median; 3 at the median; and 8 below the median.

FASTUS improved the average precision (non-interpolated) compared to the GE input on 17 of the 23 topics. On 12 of these the resulting average precision was above the median; in seven of these cases, we transformed above median input into an even better ordering. On one of these 7, topic #10001 (soil pollution), FASTUS had the best average precision (.6322). There are several possible reasons for this success. One is that there was less training data for this topic than for any of the others: 100 articles instead of (app.) 1000. Our

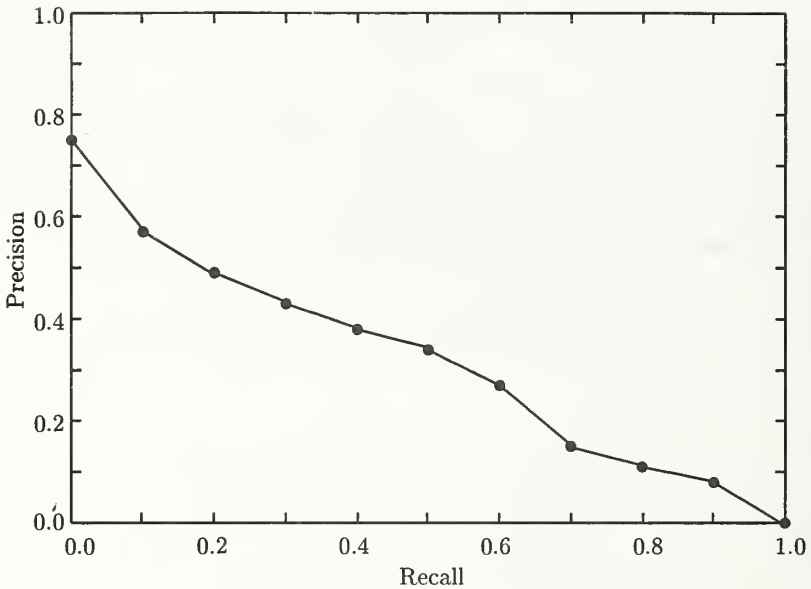


Figure 1: Recall vs. Precision for the 23 Topics

approach may suffer less from this relative scarcity of training data than purely statistical approaches. We only wrote grammars for two of the topics that had 100 or fewer training articles, so this is still conjecture. The other topic of this kind was #282 (violent juvenile crime), on which we very slightly improved above-median input. Second, we were able to reuse parts of the grammar from topic #12 (water pollution) and we benefitted in much the same way as if we had had more relevant articles. Finally, the topic may just be one where the information tends to be expressed in ways that our patterns can recognize.

On six of the topics, FASTUS lowered the average precision of the input order. A characterization of these cases is instructive. Two of these, topics #23 (legal repercussions of agrochemical use), and #194 (writer's earnings) had 7 and 8 relevant documents in the training sets, respectively. When faced with that little data, we could only guess at the various ways in which relevant information might be expressed and at which patterns would recognize them. Obviously we did not make very good guesses.

One of the six topics on which we degraded input performance was topic #11. As mentioned above, we departed from our normal method on this topic, assigning maximum scores of 250 to patterns in order to require that at least four phrases match. We did quite poorly on this topic as well.

Obviously, our approach is sensitive to the way information is expressed. Topic #228 (environmental cleanup success stories) represents an example of a topic to which our approach is not well suited. The relevant articles were not

characterized by a relatively small number of highly indicative phrase or event types—in this respect this topic was like #11—and our approach did poorly.

We have not been able to characterize our performance on the two remaining topics on which we degraded performance, beyond being convinced that we could have and should have done better.

6 Discussion and Conclusion

We have described an experiment in the use of a particular kind of Natural Language Processing technology within an Information Retrieval application. The experiment involved adapting FASTUS, an Information Extraction system, for use as a post-filter to be run over the output an IR system, GE's version of SMART. The results of the experiment are of two different kinds: First, the experiment motivated significant changes to the architecture of FASTUS, changes aimed at making it easier to develop application-specific grammars. The resultant grammars can be used for typical IE tasks, as well as for IR tasks. Second, the results on the TREC6 routing task, while certainly not impressive, just as certainly do not foreclose the possibility of using IE technology in this way in IR applications. Rather they suggest that some care must be exercised in determining the proper range of application of this mixed-technology approach to IR, for there is little reason to think it is appropriate everywhere. At least two simple guidelines can already be induced; one purely quantitative, the other, not:

- There must be sufficient data, in particular, enough relevant articles, (i) to accumulate patterns for the initial grammar-writing exercise and (ii) to use as a training corpus for adding to and "debugging" those grammars.
- There must be fairly reliable indicators of relevance that are fully phrasal in structure.

We have also learned some other more engineering-oriented lessons:

- If relevant data is scarce in a given corpus, it is worth it to go out and look for more.
- In following a hybrid approach such as ours, it is important to use the output of the IR system in training.

Perhaps these latter lessons should have been obvious from the beginning. In any event, we hope to make good use of them, and others, as we pursue this approach to applying Information Extraction technology for Information Retrieval.

7 Acknowledgments

We would like to thank Matt Caywood and Mabry Tyson for their efforts in making FASTUS run over much larger amounts of text than ever before. We would also like to thank Tomek Strzalkowski and his group at GE for providing us with their system's output. Without it we would not have been able to participate. We are happy to acknowledge that this work was funded under Contract No. N66001-94-C-6044 from the Naval Command, Control, and Ocean Surveillance Center. We also received internal research and development funding from SRI International.

References

[Appelt et al.1995] Doug Appelt, John Bear, Jerry Hobbs, David Israel, Megumi Kameyama, Andrew Kehler, Mark Stickel, and Mabry Tyson. 1995. SRI International's FASTUS System MUC-6 Test Results and Analysis. In *Proceedings of the 6th Message Understanding Conference*, ARPA, Columbia, MD.

[Appelt et al.1996] Doug Appelt, John Bear, Jerry Hobbs, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1996. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Emmanuel Roche and Yves Schabes (eds.) *Finite State Devices for Natural Language Processing.* MIT Press, Cambridge, MA.

[Berger, Pietra, and Pietra1996] Adam Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

[Croft et al.1996] Bruce Croft, James Allan, Lisa Ballestros, James Callan, and Zhihong Lu. 1996. Recent Experiments with INQUERY. In *Proceedings of TREC-4*, to appear.

[Faloutsos and Oard1996] Christos Faloutsos and Douglas Oard. 1996. A Survey of Information Retrieval and Filtering Methods. Technical Report, Information Filtering Project, University of Maryland, College Park, MD.

[Grishman and Sundheim1995] Ralph Grishman and Beth Sundheim. 1995. Design of the MUC-6 Evaluation. In *Proceedings of the 6th Message Understanding Conference*, ARPA, Columbia, MD.

[Harman1996] Donna Harman. 1996. Overview of the Fourth Text REtrieval Conference (TREC-4). In *Proceedings of TREC-4*, to appear.

[Hearst1992] Marti Hearst. 1992. Direction-Based Text Interpretation as an Information Access Refinement. In [Jacobs1992].

[Jacobs1992] Paul Jacobs (ed.) 1992. *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval.* Lawrence Erlbaum Associates, Hillsdale, NJ. ew Jersey.

[Lewis1992] David Lewis. 1992. Text Representation for Intelligent Text Retrieval: A Classification-Oriented View. In [Jacobs1992].

[Sparck Jones1992] Karen Sparck Jones. 1992. Assumptions and Issues in Text-Based Retrieval. In [Jacobs1992].

[Kehler1996] John Bear, David Israel and Andy Kehler. 1996. Using Information Extraction to Improve Document Retrieval. Unpublished mss.

Interactive information retrieval using term relationship networks.

Jim McDonald, William Ogden and Peter Foltz

New Mexico State University
Box 30001/3CRL
Las Cruces, NM 88003
{jemlogdenlpfoltz}@crl.nmsu.edu

ABSTRACT

Users have difficulty retrieving information from ad-hoc, textual databases because, by definition, they don't know precisely what's in them. Thus, users submit queries that contain the wrong terms or which don't contain enough information to retrieve all and only those documents relevant to their information needs. Our approach to these problems is to provide users with abstract representations of database content, in the form of Pathfinder networks linking related terms used in the database. This allows users to recognize and select appropriate query terms. The networks displayed to users are derived from textual analyses of documents retrieved from initial queries and, thus, the process can be thought of as a form of relevance feedback. Compared to other relevance feedback methods, however, the network displays can show important relationships between the query terms and terms suggested by the system. In the study to be reported, we compared the performance of two information retrieval systems Zprise, a control system, and ***InfoView***, a system that uses our network displays. Participants used both systems to perform an "aspectual retrieval" task using the six topics. Preliminary results from this study suggest that when participants used ***InfoView*** they took less time to identify topic aspects and were at least as successful as when they used Zprise.

Our approach to interactive information retrieval is based on a simple premise: users of information retrieval systems can't find the information they seek because they don't know exactly what's in the databases they're searching. This tautology can be expanded into a couple of more interesting claims. First, users aren't able to formulate effective queries because they don't know (or can't recall) appropriate query terms. Second, if users are provided with appropriate information about database content, they are able generate (or select) better query terms and subsequently improve retrieval performance. In other words, if users have a good understanding of database content, they do a better job of retrieving the information they seek.

One approach to improving information retrieval is relevance feedback, an inherently iterative technique (e.g., Rocchio, 1971). A user first formulates a query based on her information need and understanding of database content. She then submits the query to the information retrieval system and a list of documents (or document titles) is returned which the system deems relevant to the search request, usually ordered according to relevance. The user examines the document titles or reads the documents in order to determine which are relevant. In its simplest form, the user then modifies her original query by adding or removing terms, then resubmits the query to the information retrieval system. In this scenario query modification is based on an analysis of the retrieved

documents and involves the inclusion of terms which are synonyms or appropriate modifiers in the context of the database being searched.

Some relevance feedback techniques attempt to incorporate relevance judgments into the retrieval process by comparing relevant to non-relevant documents automatically. Even these systems, however, require users to examine retrieved documents in some detail in order to determine which are relevant. Such examination can be a time-consuming and error-prone process. Judging relevance may require users to examine a large number of documents, some which may be fairly low on the ranked list returned by the information retrieval system. On the positive side, relevance feedback has been shown to be an extremely effective technique for improving information retrieval performance (Salton &. Buckley, 1990)

Our objective is to achieve the positive results typical of relevance feedback techniques while requiring less effort on the part of the user. Thus, we provide users with information about database content in a form which minimizes time and effort while maximizing content coverage. Our technique is based on the notion that abstract representations derived from statistical analyses of text (documents) can be used to improve queries (and retrieval performance) without requiring users to examine documents or even document titles. There are two components to an information retrieval system which incorporates this approach. First, routines must be provided to analyze database content in order to produce the representations. Second, an interface must be constructed which allows users to interact with these representations in order to improve their queries. Each of these components in turn involves a number of processes, described in the following sections. The *InfoView* information retrieval system used in this study is based on the model described.

Database Analysis

The first step in producing database representations is index-term selection (indexing). In general the goal is to select representative index terms which are capable of discriminating among database topics. Although retrieval itself may require complete indexing, the index terms displayed to users are typically a subset selected to be representative of database content and capable of discriminating among topics. Once a set of index terms has been selected, pair-wise distance estimates are obtained by transforming co-occurrence data. We typically calculate the co-occurrence of index terms within sentence units and transform these data using Dice's coefficient. However, larger units (e.g., paragraphs or documents) and other transformations may be more effective. Finally, networks are produced using the Pathfinder network algorithm which effectively eliminates connections (associations) among index terms until only the strongest or most important associations remain (McDonald, Plate, & Schvaneveldt, 1990; Schvaneveldt, Durso, & Dearholt, 1989).

The User Interface

The Pathfinder algorithm takes as input a matrix of distances estimates between entities, in this case index terms, and produces as output a network specification. For the purposes of information retrieval, the nodes in these networks are index terms and the links represent the strongest

associations between the terms based on frequency of co-occurrence. In order for users to interact with these associative networks, graphical representations are produced. Because the database networks are typically very large, often consisting of thousands of nodes, they cannot be effectively displayed. In order to focus interaction within appropriate subsections of the complete network, users submit one or more index terms and portions of the complete index-term network (i.e., subnets) are selected and graphically displayed. The size of these subnets is specified as the

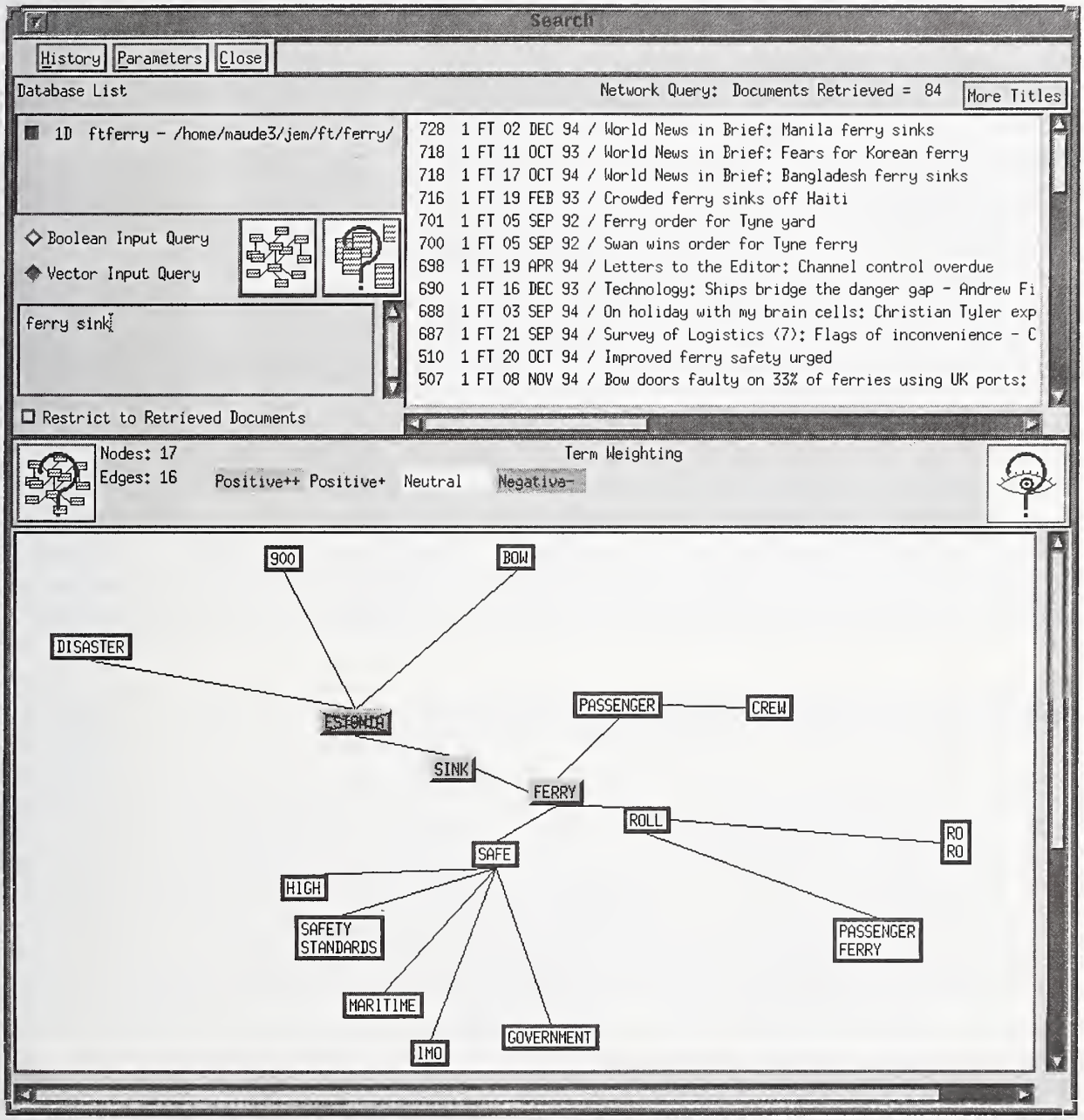


Figure 1. The *InfoView* user interface

number of nodes within a given radius (i.e., number of links) of the query terms.

The networks are used to improve information statements, or queries, by helping users to 1) substitute precise index terms for vague ones, 2) remove or "negatively weight" inappropriate index terms, or 3) add appropriate index terms. The networks can be thought of as "pictures" representing the associations in the database and are to be used to modify queries so as to do a better job of retrieving relevant documents. Users can modify their query by interacting directly with the network display. For example, Figure 1 shows the result of a user interaction with the *InfoView* system during a TREC interactive task. The topic "Ferry Sinkings" requires the user to find as many reports of ferry sinkings where 100 or more people lost their lives. The subnet show that the key terms "ferry" and "sink" are highly associated with the word "Estonia" which was the name of a ferry which sunk, killing over 900 persons thus gaining a lot of publicity. Because most of the reported ferry sinkings are referring to the "Estonia" and the user is trying to find *other* incidents, the user has used the mouse to "negatively weight" the "Estonia" term. The resulting list of document titles (displayed in descending order of estimated relevance in the top right corner of the display) shows the top four relevant documents are indeed about other ferry sinkings.

The *InfoView* model produces networks based on a complete analysis of a specified database. The subnets displayed to users are thus portions of the complete network and are based on the co-occurrences of index terms throughout the collection and across topics. We have observed that networks of this sort, derived from an analysis of the complete database, are not always optimal for information retrieval purposes. This limitation seems to stem from the fact that index terms in such networks become connected because they tend to be associated throughout the collection and across topics. However, what's needed are index terms capable of discriminating between topics, and such associations are often weakened by a global analysis. In this study we modified the *InfoView* model to produce associations based on the co-occurrence of index terms within sets of retrieved documents, rather than across the entire collection. Our TREC-6 Interactive Track effort was designed to test this approach.

TREC-6 Interactive Track Method and Results

Database preparation for the current study consisted of the following steps. First, a subset of documents from the Financial Times database were selected using a Boolean query with the terms in titles for each of the six TREC-6 Interactive Track topics. The sets of documents retrieved were treated as independent databases and Pathfinder networks were produced using the procedure outlined above. When participants were required to perform the aspectual recall task on a given topic, the network for that database was accessed by the *InfoView* system. This is a simulation of the system we propose to build and has several limitations. First, by selecting documents using Boolean search, it is quite likely that some relevant documents were excluded from the individual topic databases and hence were not available to participants, limiting recall. In the next version of our system we will use Latent Semantic Indexing (LSI) (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990) to select relevant document sets. Second, although the document sets retrieved for topics were relatively small and more homogenous than the complete collection, further improvements are hypothesized if these sets of documents are grouped or clustered in LSI space and networks formed on these topic sets.

InfoView was used as our "experimental" system in the TREC-6 interactive track according to the method specified for the track. (see http://www-nlpir.nist.gov/~over/t6i/ for the complete specification.) We had four participants each using both *InfoView* and *Zprise* (the control system). Their results were combined with track participants from the seven other interactive track sites. Like all other sites, our experimental, *InfoView* system did not significantly differ with respect to any dependent measure. In general, our participants using InfoView were faster (67 sec) with as least as good of aspectual recall as when they were using the *Zprise* control system.

Future work

Despite the inconclusiveness of the present study we are encouraged by the progress we are making toward better interactive systems and their evaluation. We are in the process of experimenting with the aspectual-recall comparison paradigm. We are currently conducting an extension of the method used in the TREC-6 Interactive track by extending the latin square design to control for task order. Further we intend to look at a new dependent measure we call "aspectual change". Aspectual change will measure the difference between what people know about a topic before and after an interactive information retrieval session. We believe this could be a very sensitive measure of the effectiveness of an information retrieval system's user interface.

References

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391-407.

McDonald, J. E., Plate, T. A., & Schvaneveldt, R. W. (1990). Using Pathfinder to extract semantic information from text. In R. Schvaneveldt (Ed.), Pathfinder associative networks: Studies in knowledge organization. (pp. 149-164). Norwood, NJ: Ablex.

Schvaneveldt, R. W., Durso, F. T., & Dearholt, D. W. (1989). Network structures in proximity data. In G. H. Bower (Ed.), The psychology of learning and motivation: Advances in research and theory (Vol. 24, pp. 249-284). New York: Academic Press.

Rocchi, Jr., J. J. (1971) Relevance Feedback in Information Retrieval. In G. Salton (Ed.) The Smart System - Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, JN, 313-323.

Salton, G & Buckley, C. (1990) Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 41(4): 288-297.

FREE RESOURCES AND ADVANCED ALIGNMENT FOR CROSS-LANGUAGE TEXT RETRIEVAL

Mark W. Davis and William C. Ogden

Computing Research Laboratory
New Mexico State University
Box 30001/3CRL
Las Cruces, NM 88003
madavis@crl.nmsu.edu

ABSTRACT

For the Cross-Language Text Retrieval Track in TREC 6, NMSU experimented with a new approach to deriving translation equivalents from parallel text databases, and also investigated performing automatic, dictionary-based translation of query terms by using a dictionary that could be queried remotely via the World Wide Web. The new approach to building bilingual translation lexicons involved aligning parallel texts at the sentence level, and then performing further alignments at the sub-sentence level. The process initially chooses alignment anchors based on N-gram matches between cognate terms. Term and phrase matching is then performed between the anchor points by finding the most direct path from one anchor to the next, penalizing larger steps over runs of terms. The collected term translations are then used as equivalents for a query translation process and the translated query is then submitted to a monolingual retrieval engine. The results are compared against the performance of a monolingual French-French retrieval system, and against a translated query formed from a very high-quality bilingual dictionary accessed directly over the World Wide Web. A combined approach is also presented that uses terminology from both the dictionary and, where the dictionary lacks coverage, supplements the query translation using terms from a parallel text database.

OVERVIEW

A Cross-Language Text Retrieval (CLTR) system retrieves documents in a language that is different from the query language. Various approaches have been proposed for CLTR. An early experiment used hand-built translation thesauri (Salton, 1971). Recent work has extended the use of lexicons to make use of bilingual machine-readable dictionaries (MRDs) of a general nature to translate query terminology (Ballesteros and Croft, 1997, Davis and Ogden, 1997, Kwok, 1997 and Hull and Grefenstette 1996). A subtle variation on these methods is to use controlled-vocabulary thesauri or other specialized resources for query translation, which often provide excellent coverage of highly technical subject matter.

A dramatic alternative to the use of prepared bilingual or multilingual lexicons is to rely on the information contained in parallel texts (texts that are translations of each other) to train or derive a translation model. One approach, Latent Semantic Indexing (LSI), maps documents into a reduced-dimensionality space, based purely on term-document co-occurrence statistics in parallel

texts (Dumais, Landaurer and Littman, 1995). Queries can, in turn, be mapped into the same space and the nearest documents to the query returned. In LSI, queries can be expressed in any of the shared languages of the training texts. Alternative methods include training linear, but underdetermined, translation models using an iterative least-squares minimization (Dunning and Davis, 1993), and applying stochastic optimization approaches to try to match query performance in both languages over a parallel training text set (Davis and Dunning, 1996). Approaches based on these methods have shown moderate promise, although no large-scale implementation has yet demonstrated performance that matches hybrid methods or methods that rely exclusively on MRDs for the translation task.

Despite some early experimental successes, the task of Cross-Language Text Retrieval remains dauntingly difficult, if only for the reason that resources for translation remain exceedingly expensive. Even for a system that shows startling performance in one language pair, moving to a new language pair often requires completely new resources and personnel. Parallel text is the relatively rare by-product of large-scale translation operations, while bilingual MRDs are the tightly-held intellectual property of dictionary companies, commanding impressive royalty fees for widespread application. Further, tuned lexicons for machine translation applications remain the most closely guarded inner secrets of machine translation companies. A third alternative, "comparable texts", which are matched according to topic, but not necessarily direct translations, are also plausible resources for extracting query translation terminology, but are not clearly easier to amass than true parallel text.

At NMSU's Computing Research Laboratory, we have found that the problems associated with the lack of good resources for translation are rapidly being offset by the increased availability of materials on the World Wide Web (WWW). Our approach for the TREC-6 Cross-Language Retrieval Track only uses freely available WWW resources to translate English queries into French, using a combination of new text alignment techniques for parallel text WWW resources, and bilingual MRDs. The resulting French queries are then submitted to a monolingual retrieval engine to retrieve French documents. The resulting documents could then be translated or glossed back into English using the same resources combined in an approach like that presented in Davis and Ogden (1997). Our TREC-6 submission continues to emphasize our commitment to one very practical scenario: a monolingual information retrieval user who submits a query against a collection of documents in another language, and who will then need translation aids to assess the relevance of the retrieved documents.

IS THERE A FREE LUNCH?

On-line bilingual dictionaries represent a powerful new opportunity for research and development of CLTR technology. Using a list of morphological root forms for terms in a large English text collection, custom Web robots can acquire on-line resources like bilingual resources for use in CLTR tasks. We have recently developed robots to do exactly that and have acquired reasonable "kernel" bilingual dictionaries from English to ten other languages. The number of headwords available for each language pair is small in comparison with printed dictionaries, but can be

quickly expanded by a user with access to corpus analysis tools.

Languages	*Headwords*
English-Afrikaans	3,733
English-Dutch	9,853
English-Danish	3,715
English-Finnish	2,832
English-French	3,582
English-Japanese	176,528
English-Hungarian	2,479
English-Italian	2,912
English-Portuguese	2,637
English-Spanish	5,201

Table 1 Headwords for bilingual dictionaries

A comparison of the coverage of a large corpus by a kernel dictionary like the English-Spanish dictionary in the table, above, to a larger print dictionary is revealing. For a collection of 10.7 million words (TREC Spanish AFP collection) and 207,433 unique words filtered by a 30,805 word Collins bilingual dictionary headword list, case-normalized and stemmed in IR fashion (Davis, 1996), 187,103 words remain (90.2%). The 5,201-word dictionary leaves 204,227 words (98.5%).

An analysis of a randomly drawn 100 words from the unaccounted-for segment using the 30,805 word Collins dictionary indicates that 11 were abbreviations, 9 were foreign words, 49 were proper names and 31 were other words. If this pattern is representative of the collection as a whole, then abbreviations, proper names and foreign words represent a startling 69% of untranslatable words. Some abbreviations between Latin and Germanic languages go straight across (km), but some do not (NATO and OTAN), and most proper names go directly across, or require only minor accent normalization rules to account for. This pattern will not hold for translations between radically different language pairs, however, and we can expect that as CLTR is expanded to handle distally-related language pairs that the impact of these terms will grow as well.

The promise of using parallel corpora to compensate for the narrow view of dictionaries does not appear to present dramatically wider coverage of a corpus. The Spanish parallel document set from Pan American Health Organization (PAHO), consisting of 22,094 unique words drawn from 94,313 total words, leaves 201,660 words (97.2%) when filtering the same AFP document set. Making good use of parallel corpora for translation is imperfect at best, however, so the potential value of even this meager amount of coverage to CLTR applications remains suspect.

It seems that if there is a free lunch for CLTR due to free resources, then it is primarily due to the limited coverage provided by *any* translation resource, and the significant impact that direct matching of proper names and abbreviations has on retrieval performance for specialized queries. This benefit will likely disappear when, say, applying English queries to Chinese databases, without significant work for developing extensive proper name databases, or so-called *onomastica*, or

for providing the ability to transliterate proper name expressions into the target language.

TREC 6 AND THE CLTR TRACK

Our experiments for the CLTR track of TREC 6 involved several freely available resources. First we remotely queried a large English-French bilingual dictionary at the University of Chicago to obtain translations of English query terms. We then supplemented the dictionary translation with additional terms derived from a parallel text database created using phrase-level alignment. All of the cross-language studies used only the French description field of the queries, a short statement of the topic.

The University of Chicago dictionary was created for use in a machine translation project and was therefore fairly clean, requiring minimal filtering to extract the key equivalent set. The dictionary contained entries for 209 out of 257 English terms (81.3%), with notable omissions including:

acupuncture
AIDS
resurgence
worldwide
franchise
pollution
Berlin
labor

Table 2 Key terms not covered by French-English dictionary

Not being able to translate AIDS in topic 7 presented perhaps the most serious deficiency for the system. Although "acupuncture" and "Berlin" translate straight across, AIDS does not. Our hypothesis was that for a high-quality bilingual dictionary like this one, the most pressing need was to improve the dictionary's coverage for terms or phrases that were not in the original dictionary.

ALIGNMENT AND PHRASE EXTRACTION

To supplement the dictionary, we used parallel French-English parliamentary proceedings acquired automatically from the Canadian government archives. The English document set contained 51,732 words, while the French set contained 52,281 words. The documents were first aligned at the sentence and sentence-pair level using the statistical alignment procedure reported

in Davis, Dunning and Ogden (1995), and which has been used to align Spanish and English document sets for past TREC experiments (Davis and Ogden, 1997). The second part of the alignment procedure involved discovering phrase and word matches between the aligned blocks at the sub-sentence level. Unlike the methods reported in Gale and Church (1991) in which the statistics are concerned only with the relative rates of co-occurrence between terms in aligned blocks, our approach emphasized the order of text within and between the French and English blocks. In this respect, our methodology perhaps most closely resembles the methods of Melamed (1996), but uses n-gram matching between cognate terms within that ordered set as the primary feature for establishing anchors. Once these anchors are established, the regions between the anchors are analyzed using a phrase-finder tuned for English and French to extract significant matching phrasal terminology to extend the bilingual dictionary. This procedure resulted in a dictionary of 7,869 pairs, including "sida" translates to "AIDS".

RESULTS

The CLTR results demonstrated extremely wide variance between the performance of the best and worst groups. A preliminary comparison between the CRL cross-language runs and all submissions shows CRL performing at or above the median on 13 out of 21 judged topics, with the cross-language system turning in the best performance of all systems on one topic, and among the worst on one other topic. Although in the preliminary results we saw cross-language results that were superior to the monolingual case, in the revised results the best of the cross-language runs were slightly poorer than the cross-language case. The results are graphed in Figure 1 on Page 6.

Run	*Average Precision-Recall (all queries)*	*% of monolingual case*
Monolingual French	0.1484	—
English-French using unmodified English-French dictionary	0.1371	92.38%
English-French using parallel-corpus derived bilingual lexicon	0.0357	24.06%
English-French using augmented dictionary	0.1428	96.23%

Table 3 Performance of CLTR approaches

The consequences of augmenting the dictionary using parallel texts are especially evident in Query 7, which contained the term AIDS. AIDS was not in the dictionary but was successfully added by the parallel text process. The average precision-recall for Query 7 went from .0095 to .1296 due to the addition of the discovered term. Indeed, all of the performance difference

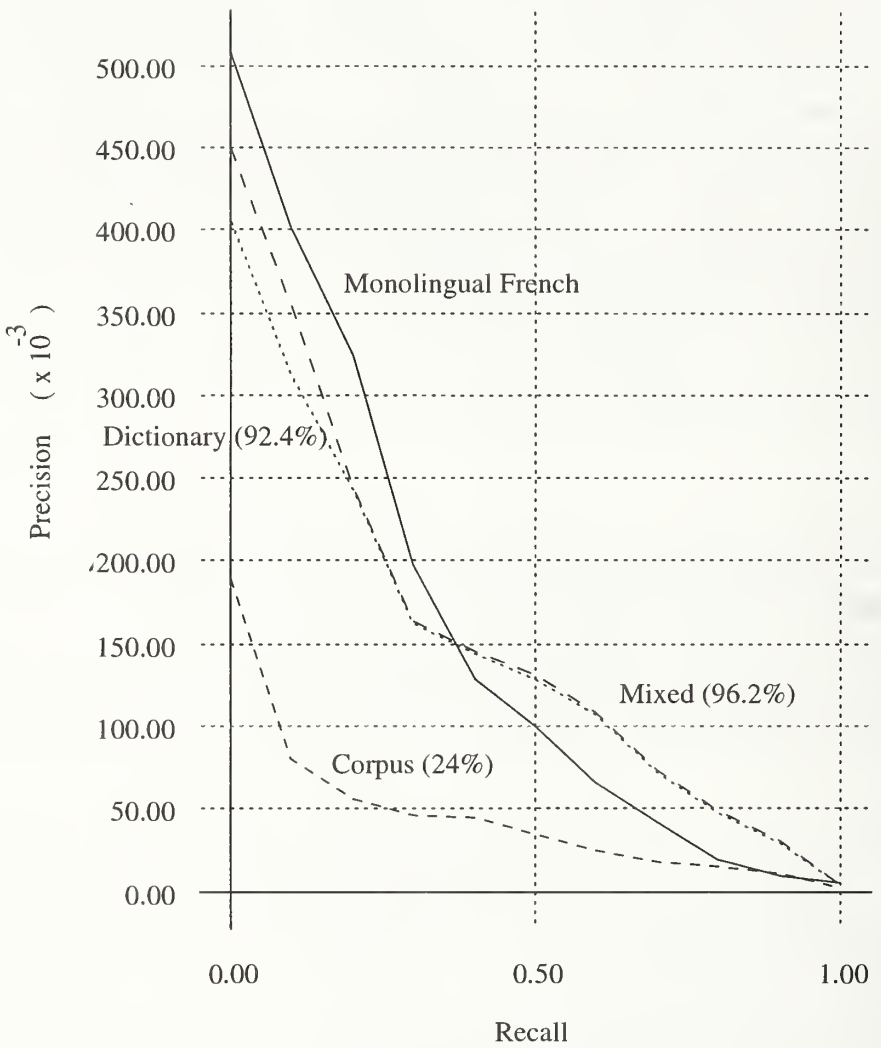


Figure 1. Average precision-recall curves for monolingual, dictionary and corpus-based cross-language retrieval techniques, as well as a method that combines the two approaches.

between the unmodified dictionary and augmented dictionary runs can be attributed to the improved performance on Query 7.

Of special interest is the comparatively poor perfomance of the corpus-based methods alone. These results bear a close similarity to those of other approaches both in TREC7 and TREC6 (Davis, 1996). The general conclusion is that extracting information and term equivalences from parallel corpora is a noisy process that results in highly noisy translation resources. The best role of such resources seems to be as a supplement to hand-prepared dictionaries, rather than as a substitute for them. Note also that extraction methods that use high-frequency term filtering to reduce error rates in turn substantially reduce the coverage of the derived dictionary to phrases and terms

that already occur in hand-prepared, general bilingual resources.

CONCLUSIONS

Cross-Language Text Retrieval is a difficult problem that is compounded by the need for rare resources like parallel texts and high-quality bilingual dictionaries. Our experiments showed that a CLTR system can be successfully built using only freely-available translation resources captured from the World Wide Web.

REFERENCE

Ballesteros, L. and W. B. Croft (1997) "Phrasal Translation and Query Expansion Techniques for Cross-Language Information Retrieval," in SIGIR '97, Philadelphia, PA, July 27-31.

Davis, M. W. and Ogden, W.C. (1997) "QUILT: Implementing a Large-Scale Cross-Language Text Retrieval System," in SIGIR '97, Philadelphia, PA, July 27-31.

Davis, M. W. (1996) New Experiments in Cross-Language Text Retrieval at New Mexico State University's Computing Research Laboratory. In *Proceedings of the Fifth Text Retrieval Evaluation Conference*, Gaithersburg, MD, National Institute of Standards and Technology.

Davis, M. W. and T. Dunning (1996) Query Translation Using Evolutionary Programming for Multi-Lingual Information Retrieval II. In *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, San Diego, Evolutionary Programming Society.

Davis, M. W., T. Dunning, and W. Ogden (1995) Text Alignment in the Real World: Improving Alignments of Noisy Translations Using Common Lexical Features, String Matching Strategies and N-Gram Comparisons. In *Proceedings of the Conference of the European Chapter of the Association of Computational Linguistics*. University College Dublin. March.

Dunning, T. E., and M. Davis (1993) Multi-Lingual Information Retrieval. *Memoranda in Computer and Cognitive Science*, MCCS-93-252, Computing Research Laboratory, New Mexico State University.

Dumais, S. T., T. Landaurer and M. Littman (1995) Automatic Cross-Linguistic Information Retrieval Using Latent Semantic Indexing. In *Proceedings of the Workshop on Cross-Linguistic Information Retrieval*, SIGIR'96, Zurich.

Gale, W. A. and K. W. Church (1991) A Program for Aligning Sentences in Bilingual Corpora. In *Proceedings of the 29th Annual Conference of the Association of Computational Linguistics,* 177-184, Berkeley, CA.

Hull, D. and Grefenstette, G. (1996) "Experiments in Cross-linguistic Information Retrieval" in *SIGIR96*, August, Zurich, CH.

Kwok, K. L., (1997) "Evaluation of an English-Chinese Cross-Lingual Retrieval Experiment," in Working Notes of the Cross-Language Text and Speech Retrieval Spring Symposium, AAAI-97 Spring Symposium, March 24-26, Stanford, CA.

Salton, G. (1971) "Automatic Processing of Foreign Language Documents," in *The Smart Retrieval System*, ed. Salton, G., Prentice-Hall, Englewood Cliffs, NJ.

APPENDIX 1: CLTR Questionnaire

To those of you in the CLIR track who are new to TREC, this questionnaire makes a distinction between "topics", the descriptions furnished by NIST, and "queries", the actual text submitted to your retrieval system for searching.

Queries may simply be a copy of some part or all of the topic, may be derived automatically from the topic, or may be formulated manually based on the topic description.

1. OVERALL APPROACH:

1.1 What basic approach do you take to cross-language retrieval?

[X] Query Translation

[ ] Document Translation

[ ] Other:

1.2 Were manual translations of the original NIST topics used as a starting point for any of your cross-language runs?

[ ] No

[X] Yes: The NIST-supplied English topics.

1.3 Were the automatically translated (Logos MT) documents used for any of your cross-language runs?

[X] No

[ ] Yes:

1.4 Were the automatically translated (Logos MT) topics used for any of your cross-language runs?

[X] No

[ ] Yes:

2. MANUAL QUERY FORMULATION:

2.1 If query formulation involved manual effort, how fluent was the user in the source (query) language?

[ ]

2.2 If query formulation involved manual effort, how fluent was the user in the target (document) language?

[ ]

3. USE OF MANUALLY GENERATED DATA RESOURCES:

3.1 What kind of manually generated data resources were used?

[X] Dictionaries

[ ] Thesauri

[ ] Part-of-speech Lists

[ ] Other:

3.2 Were they generated with information retrieval in mind or were they taken from related fields?

[ ] Information Retrieval
[X] Machine Translation
[ ] Linguistic Research
[ ] General Purpose Dictionaries
[ ] Other:

3.3 Were they specifically tuned for the data being searched (ie. with special terminology) or general-purpose?

[ ] Tuned for data; Please specify:
[X] General purpose

3.4 What amount of work was involved in adapting them for use in your information retrieval system.

[ ] None
[X]: robotic retrieval from WWW, filtering to eliminate duplicate headwords

3.5 Size

[X] 3582 entries
[ ] _ MBytes

3.6 Availability? - Please also provide sources/references!

[ ] Commercial
[ ] Proprietary
[X] Free: www.travlang.com and www.uchicago.humanities.edu
[ ] Other:

4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES:

4.1 Form of the automatically constructed data resources?

[X] Lexicon
[ ] Thesaurus
[ ] Similarity matrix
[ ] Other:

4.2 What sort of training data was used to construct them?

[ ] Same data as used for searches:
[ ] Similar data as used for searches:
[X] Other data:

4.3 Size

[ ] _ entries
[ ] _ MBytes

4.4 Was there any manual clean-up involved in the construction process?

[ ] Yes:
[ ] No

4.5 Rough resource estimates for building the data resources (ie. an indicator of the computational complexity of the process).

[X] 0.1 hours
[ ] _ MBytes of memory used
[ ] _ temporary disk space

5. GENERAL

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?

[ ] Very dependent, _
[ ] Somewhat dependent,
[X] Easily replacable, _
[ ] Don't know

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?

[ ] Yes, a lot, _
[X] Yes, somewhat: see estimates of data coverage in
[ ] No, not significantly, _
[ ] Don't know

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?

[ ] Yes a lot, _
[X] Yes, somewhat: see estimates in See "IS THERE A FREE LUNCH?" on page 2.
[ ] No, not significantly,_
[ ] Don't know

5.4 Are similar resources available for other languages than those used?

[X] Yes, French, Italian, Portugese, Japanese, Hungarian, Aftikaans, Dutch, Danish, Finnish
[ ] No

EMIR at the CLIR track of TREC6

Faiza Elkateb and Christian Fluhr

Commissariat à l'Energie Atomique
DIST
CEA/Saclay
91191 Gif/Yvette cedex
France

E-mail : fluhr@tabarly.saclay.cea.fr

1 Introduction :

EMIR (European multilingual information retrieval) is a European ESPRIT project whose aim was to demonstrate the feasibility of a crosslingual interrogation of fulltext databases based on a general multilingual interrogation. The project lasted from November 90 to April 94. A part of the results are included into a commercial product "SPIRIT" released by the T.GID company in France.

2 Principles of EMIR used in this experiment :

Remark : only a part of the EMIR results have been used in this experiment.

Linguistic processing :

Linguistic processing is done both on texts and on queries. It is a morphosyntatic processing whose aim is to identify and normalize the concepts inside the documents and the queries. Normalized concepts can be single words, idiomatic expressions or compounds (in this experiments only couples) recognized by the fact that their components are in dependency relation. The normalized words are tagged by their part of speech.

In this experiment, only contiguous idiomatic expressions have been taken into account. That means that for example , in English, verbs with a post position that can be non contiguous are considered as two separate words.

Weighting :

The weighting of documents is done by the computation of a weight for each normalized words according to the fact that they are more or less discriminent. This word weight is used to compute a weight for each intersection query documents. All document having the same intersection with the query have the same weight. They are grouped into a class of intersection which is characterized by the "best" boolean query that can be used to obtain these documents from the original query words.

In this experiment we have not used the possibility of excluding a documents from a class of intersection if the words defining the class are not all in the best informative page. That means that some long documents which do not contain all the query words concentrated in one or more parts of the document can be in a better position in the ordered list of answers than expected.

Reformulation :

The reformulation tool is used to infer from the original query words new words expressing the same concepts and that can be found into the texts. It can be both used in monolingual interrogation and in crosslingual interrogation. Inferences are conditioned by the part of speech. For example "light" adjective infers "léger" adjectif in French and not "lumière" noun.

In this experiments only one step of reformulation have been used for crosslingual interrogation (English to French reformulation). The monolingual reformulation (word of the same family and synonyms) for the French to French interrogation has been used.

These reformulation dictionaries are general purpose dictionaries. They have not been modified for the experiment.

All possible translations are tried they are filtered by the database lexicon and by the cooccurrences of translations in the best documents.

Non feedback on the translations have been applied for the crosslingual interrogation.

The volumes of dictionaries are the following :

French monolingual (single words) : 365 534 forms + 150 328 entries for automatic correction
English monolingual (single words) : 98 565 forms

French monolingual reformulation : 28 713 (lemmas)

English to French bilingual reformulation : 32109 entries (lemmas)

3 Conditions of experimentation on CLIR track data :

We had a very short time to devote to the CLIR experiment that is the reason why we have spend only the last 2 days before the deadline. The system was used without any actions on both the queries and the texts even if we have found by a detection of unknown words that there is a lot of errors.

So, no errors were corrected both in the queries and in the texts and no new words were added. That means that there was no modifications on the French monolingual dictionary used to index the French database and query in French and on the English monolingual dictionary used for English queries.

There was also no modification on the monolingual (French to French) reformulation and on the English to French reformulation.

Because of a problem on one of our disks it was not possible to generate the full French database. The interrogation has been done on the 71464 first documents out of 113 656 in the full database. (documents from 8.12/89 to 1990 have not been taken into account). This have decreased the level of the curve precision-recall on his right part.

Construction of Queries :

Queries have been automatically build from the topics by taking the "Desc" field. The interrogation was done by a fully automatic process that take each query, send it to the system and put the answer in the suitable form for TREC-EVAL. The query is compared to the 2 textual parts of the database : title and text.

4 Example of interrogation :

To illustrate the system functioning we will give an example of monolingual and bilingual interrogation.

French query to French database SDA FF # 19: **La consommation de vin augmente ou diminue-t-elle dans le monde?**

Monolingual reformulation :

vin ===> oenologie, oenologue

augmenter ===> augmentation

diminuer ===> réduire, restreindre, amoindrir, décroître, amenuiser

monde ===> mondial, mondialement, mondialisme

Interrogation results :

Classes of intersection in a decreasing order of relevance

```
Class # 1  consommation de vin AND diminue
BSF.890612.0084
BSF.880823.0063 <=== Relevant

Class # 2  consommation de vin AND augmente
BSF.890612.0091 <=== Relevant
BSF.890609.0076 <=== Relevant
BSF.880322.0061
BSF.890825.0086 <=== Relevant

Class # 3  consommation de vin AND monde
BSF.880920.0063 <=== Relevant
BSF.890320.0052

Class # 4  consommationde vin
BSF.890612.0068
BSF.881227.0015
BSF.890404.0175
BSF.881026.0008
BSF.880823.0033 <=== Relevant
BSF.880916.0121 <=== Relevant
BSF.880502.0080
```

```
BSF.880610.0071
BSF.880328.0124
BSF.880329.0017 <=== Relevant

Class # 5  vin AND diminue AND consommation AND augmente
BSF.880607.0155
BSF.891002.0033

Class # 6  vin AND consommation AND augmente AND monde
BSF.880302.0014 þ
BSF.891015.0027 þ
BSF.881011.0056 þ
```

English query to French database SDA EF #19: **Is wine consumption/production rising or decreasing world-wide?**

Bilingual reformulation :

```
wine ===> vin

consumption ===> consommation

production ===> mise en scène, production, mise en onde, réalisation,
pièce, oeuvre, présentation, fabrication, rendement

rising ===> ajournement, clôture de séance, lever, hausse, augmentation,
résurrection, élévation, soulèvement, insurrection

decreasing ===>
=======unknown word (not in the transfer dictionary)
```

Bilingual interrogation results :

Classes of intersection in a decreasing order of relevance

```
Class # 1  wine consumption AND production AND rising
BSF.890612.0091 <=== Relevant
BSF.890609.0076 <=== Relevant
BSF.880322.0061
BSF.890825.0086 <=== Relevant

Class # 2  wine consumption AND production
BSF.880328.0124
BSF.881026.0008
BSF.880823.0063 <=== Relevant
BSF.880916.0121 <=== Relevant
BSF.880610.0071

Class # 3  wine consumption AND rising
BSF.890612.0084
BSF.890320.0052
BSF.880329.0017 <=== Relevant
BSF.890612.0068 <=== Relevant

Class # 4  production rising AND wine
BSF.890802.0014

Class # 5  production rising AND consumption
BSF.880913.0025
BSF.890906.0048
BSF.890419.0190
```

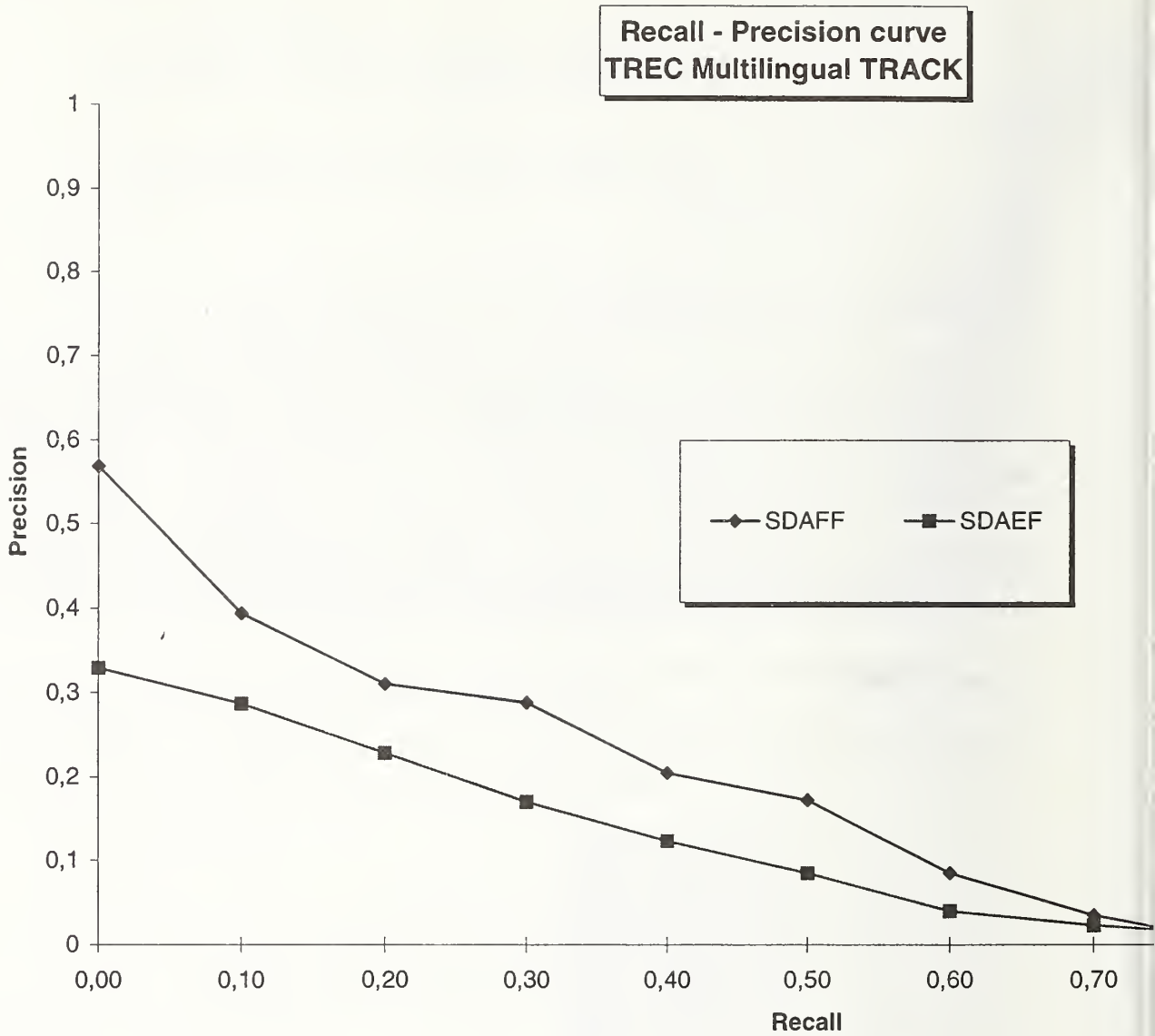
```
BSF.890420.0073
BSF.880516.0122
BSF.880329.0079
BSF.890425.0054
BSF.890620.0089
BSF.880420.0028
BSF.891031.0051
BSF.890705.0065

Class # 6  wine consumption
BSF.880502.0080
BSF.890404.0175
BSF.880823.0033 <=== Relevant
BSF.880920.0063 <=== Relevant
```

Theses examples show clearly that the functioning of the system can be easily explained to a user. The reformulation can be explained because all inferences are explicit and the user can (if he wants) interact to modify what is inferred. It can also understand why documents are considered by the system as relevant because each class of intersection is characterized by the best boolean query to get these documents The user is not obliged to follow the order suggested by the system, he has in general a sufficient information to view documents in a modified order. At end during the browsing of documents best informative pages can be directly displayed and all query and inferred words are highlighted.

5 General Results :

The results obtained on 2/3 of the database (but judged on the totality) are the following

	Fr-Fr	En-Fr
0	0,5687	0,3289
0,1	0,3936	0,2866
0,2	0,3106	0,2282
0,3	0,2882	0,17
0,4	0,2046	0,1231
0,5	0,1719	0,0852
0,6	0,085	0,04
0,7	0,0349	0,023
0,8	0,0033	0,0117
0,9	0	0
1	0	0



6 Recommandations for improvements :

The first way to improve the quality of the result is to correct errors in the queries. Then add new words of the query into the monolingual dictionaries. Of course, it is also interesting to add new words from the texts but the number of errors is so important that the work to separate new words from errors in the reject word file in very important.

After that, words in the query which have no translation in the target language must be added. For example, there was no translation in our bilingual dictionary for Moyen-Orient <--> Middle-East or ours en peluche←→ teddy bear

Monolingual reformulation can be updated with the new words from the queries and the texts

Without changing the comparison method the results can be easily increased.

Then it will be interesting to test improvement like :

In crosslingual reformulation a second step for monolingual target reformulation

In crosslingual reformulation a feedback based on the best documents to find the best translations

Use of the information on incompatible translations

Use of the new idiomatic expression recognizer developed in the framework of EMIR and that can take into account expressions that can be non contiguous and that can contain various forms of the same lemma.

Quality control on the dictionaries (consistency, exhaustivity for the parts of speech in the monolingual, exhaustivity of translations in the bilingual dictionaries)

7 Conclusion :

In an approach based on bilingual manually constructed dictionaries, the consistency and exhaustivity of the linguistic data **is essential** for the quality of the results. After that, improvements can be done using a better choice of translations by the use of the database as a semantic filter (translation feed back).

8 Bibliography :

Debili F., Fluhr C., Radasoa P., About reformulation in fulltext IRS, Conference RIAO 88, MIT Cambridge, mars 1988, A modified text has been published in "Information processing and management" Vol. 25, N° 6 1989, pp 647-657.

Fluhr C., Multilingual Information, Pacific Rim International Conference on Artificial Intelligence (PRICAI), "AI and Large-Scale Information", Nagoya, 14-16 November 1990.

Fluhr C., Radwan Kh., Fulltext databases as lexical semantic knowledge for multilingual interrogation and machine translation, EWAIC'93 Conference, Moscow, 7-9 september 1993.

Fluhr C., Mordini P., Moulin A., Stegentritt E., EMIR Final report, ESPRIT project 5312, DG III, Commission of the European Union, october 1994

Fluhr C., Schmit D., Ortet P., Elkateb F., Gurtner K., Semenova V., Distributed multilingual information retrieval, MULSAIC Workshop, ECAI96 Conference, Budapest, 12-16 Août 1996

Fluhr C., Schmit D., Ortet P., Elkateb F., Gurtner K., Distributed crosslingual indexing and retrieval engine, INET'97, Kuala Lumpur, june 1997

Radwan Kh., Fluhr C., Textual database lexicon used as a filter to resolve semantic ambiguity, application on multilingual information retrieval, 4th annual symposium on document analysis and information retrieval, las vegas, 24-26 avril 1995.

Conceptual Indexing Using Thematic Representation of Texts

Boris V. Dobrov, Natalia V. Loukachevitch, Tatyana N. Yudina

Center for Information Research
339, Scientific Research Computer Center of Moscow State University
Vorobyevy Gory, Moscow, Russia
cir@online.ru

Abstract

We present the thesaurus-based indexing technology developed by the Center for Information Research under the Information System RUSSIA project. The technology is based on using basic properties of coherent text. Initially the technology was applied for automatic processing of Russian official (government) texts. Currently the instrument is adapted to process English texts for TREC-6 routing task.

1. Introduction

The indexing approach described here is the result of NLP-technology developed under the Information System RUSSIA project. The IS RUSSIA project pursues three main goals:

- to create and support a public domain computer-based library, designed and developed to also serve as a database for social studies and university education;
- to realize NLP technology for the Russian language; and
- to develop an adequate complex of searching tools and a user-friendly English interface in order to serve as a bilingual information resource available on-line for foreign users.

The technological approach realized under the IS RUSSIA project is based on research in linguistics. It is aimed at automatic Russian language text processing, understanding and information analysis (Yudina T., Dorsey P. 1995). The main approach is to analyze the content of a text. Currently a deep-structured search image is created for every text. In addition to traditional bibliographic fields, the search image also includes thesaurus-based components: subject headings, a list of topics described, main and specific thematic nodes, mentioned descriptors, and relations between topics. The thesaurus-based components provide for thematic representation of a text that is used for indexing, categorization and summarization of a text. Ranked query results presentation is based on this technique.

The technology is currently applied to the Russian official document electronic text corpora - one of the most complicated ones. The next text corpora it will be applied to is the news media. The system will provide automatic processing, indexing, and event categorization of messages and electronic editions of Russian leading informational agencies, newspapers and magazines. In the future we hope to realize a reference technique that will expand the analytical component of the system and enable it to keep track of a situation in its dynamics, as a next step - to compare reports coming from different sources.

The technology has made it possible to develop the IS RUSSIA as an integrated information warehouse that can be searched and retrieved across in its entirety. The search engine includes a system of subject headings and thesaurus-based retrieval as well as context search techniques. The most sophisticated instrument is the Thesaurus on Sociopolitical Life. Developed as part of the project, it incorporates more than 21,000 terms and 8,000 geographic names. It assists in navigation across the huge masses of textual data and enables query expansion based on the concept relationship encoded in the thesaurus.

The IS RUSSIA has been designed as part of the international information structure so as to serve not only Russian researchers but foreign specialists on Russia as well. This application required the creation of a set of bilingual tools including a user-friendly interface, help screens, reference databases, and search instruments. The search tools include the

"System of Subject Headings" (about 200 entries) and the bilingual version of the Thesaurus on Sociopolitical Life (currently with more than 10,000 equivalents), the work is underway to translate it in full and to compose the thesaurus in English (currently it is still a set of translations from Russian). The English translation is being done in concordance with the "System of Subject Headings" of the Library of Congress, the "Legislative Indexing Vocabulary" of the Congressional Research Service (LIV 1990), the United Nations thesaurus (UNBARS Thesaurus 1976), the LegiSlate thesaurus, the Westlaw thesaurus, and the EVROVOC (thesaurus of the European Economic Community).

The search tools include the optional use of subject headings systems that are mostly popular among foreign experts (those of the Congressional Research Service, US Library of Congress and the LegiSlate; the work underway is to include the system of subject headings of the European Economic Community).

The IS RUSSIA is being developed using Oracle Server (SCO UNIX). Linguistic software mainly run under MS-DOS. Four P/5-133 were used for the TREC-6 routing task.

The IS RUSSIA integrates a wide variety of official data and documents (laws, presidential edicts and directives, governmental enactments, acts and regulations), and exceeds 150 Mb of pure document text. The collection covers the period from 1994 till now. It is updated on a regular basis from official first-hand sources, and contains all open official documents. The system includes reference data on the Russian political system (brief history, prerogatives, structure and personnel of federal institutions, political parties, churches, etc.); extended reference information on the constituent members of the Russian Federation; economic indicators and election statistics.

The team of developers is a non-commercial organization - the Center for Information Research housed at the Research Computer Center of the Moscow State University. The team includes 20 specialists from academic institutions and universities of Moscow, and consists of system analysts, programmers, linguistic researchers and social scientists. Financial support of the project was provided by foreign charitable funds, the Russian government, and scientific funds.

The IS RUSSIA was initially designed to serve as an information warehouse for social investigations. This purpose requires a representative and regular updated complex of databases storing data and documents from a wide scope of resources. The Internet-based foreign resources may significantly enrich the information flow. Special part of the IS RUSSIA project are efforts aimed at applying the developed NLP technology on processing of large collections of English texts. TREC-6 is our first experience using English texts.

2. Thematic Representation of Text

The core of the indexing technology is the thematic representation of a text. The thematic representation serves for description of contents of a document and is constructed using thesaurus knowledge about terms and property of text cohesion.

Text cohesion is achieved through semantically related terms, reference, ellipsis and conjunctions (Halliday and Hasan, 1976). Lexical cohesion is the most frequent type of cohesion. It can be achieved by repetitions, synonyms and hyponyms (reiteration) or by thematically related terms (collocation) for example: *aircompany*, *aircraft*.

Sequences of terms which the lexical cohesion relation holds can be incorporated into lexical chains. It is clear intuitively that lexical chains are connected with discourse and topical structure of the text, and so their recognition is very important for automatic text processing and representation of document content. To construct lexical chains, a linguistic resource describing relations between terms is needed. Both (Barzilay and Elhadad, 1997) and (Hirst and St-Onge, 1997) construct lexical chains based on WordNet (Miller et al. 1990). However, WordNet does not describe thematic relations between synsets (Climent et al. 1996) and therefore thematic relations are not used in the constructions of lexical chains.

Consideration of thematic relations changes a system of lexical cohesion relations in the text because a term can support some lexical chains simultaneously. For example, *minister* can support lexical chains of *government* and *ministry*, *astronaut -- cosmonautics* and *human*, *ratification* - lexical chains of *international treaty*, *the State Duma of Russia* and *the Congress of the USA* at the same time. It means that lexical cohesion is not based on a set of isolated lexical chains but on a complicated net of different relations between terms.

Semantically or thematically related terms of the text are not always connected with lexical cohesion relation. The existence of this relation is more likely for related terms in the same segment of the text than for terms in different segments and for domain specific terms than for words of common language. At the same time lexical cohesion can be the only means connecting text segments situated far from each other in the text. Thus it can be difficult to automatically decide if the relation of lexical cohesion holds between two related terms of the text.

In (Barzilay and Elhadad, 1997) the terms in text segments can be incorporated into lexical chains if they are members of synset of WordNet, if one is the child of the other in the hyperonym graph and in some cases if they are siblings in the hyperonym graph. Two lexical chains from different text segments are incorporated into a single chain if they contain a common word with the same sense. The lexical chains constructed in this manner can include terms that are not related to each other and have a bizarre form if they are represented as graphs of concepts.

We require that a lexical chain must represent a concept from the topical structure of text. Van Dejk (van Dejk, 1983) describes the topical structure of text - the macrostructure- as a hierarchical structure in the sense that the theme of a whole text can be identified and summed up to a single macroproposition. The theme of the text can usually be described in terms of less general themes which in turn can be characterized in terms of even more specific themes, and so on.

We approximate the highest macroproposition of the macrostructure with the set of macroconcepts that name the predicate of the macroproposition and its arguments. Each text is mainly devoted to description of the relations between these macroconcepts. This means (and our experiments confirm (Lukashevich, 1995)) that in most cases repetitions and synonyms of a macroconcept in the text are co-referent or are in relation of conceptual identity with the macroconcept. In most cases hyponyms, hyperonyms and thematically related terms of the macroconcept participating in subtopics of the text characterize different aspects of this macroconcept. Thus we can construct a lexical chain including a macroconcept and all text terms related to the macroconcept. We call such lexical chains 'thematic nodes'. The term that all terms of the thematic node are related to is called 'thematic center'.

Since we could construct thematic nodes for any term of the text as a thematic center, the question is how to distinguish thematic nodes of macroconcepts (main thematic nodes) from all possible thematic nodes of the text. Again we must remember that the text is devoted to description of relations between macroconcepts and so most sentences of the text must characterize these relations. This means that elements of different main thematic nodes occur together in sentences of the text more often than other terms. This distinguishes main thematic nodes from all other thematic nodes for texts of any size and different genres.

Thus the thematic representation of text is a hierarchical structure of terms where terms semantically or thematically related to thematic centers are gathered in thematic nodes. Thematic nodes whose thematic centers can characterize contents of the text are called main thematic nodes. The thematic representation hierarchy characterizes the importance of terms in the text: the thematic center is more important than other terms of the thematic node, and terms of main thematic nodes are more important than terms of other thematic nodes.

Thematic representations are created on the basis of detailed description of the domain, represented as a thesaurus. Our Thesaurus was specially created as a tool for automatic processing of texts in the broad domain of sociopolitical life and has some essential distinctions from conventional thesauri created for manual indexing.

3. Thesaurus on Sociopolitical Life

We created our Thesaurus as a tool for automatic indexing -- the Thesaurus on Sociopolitical Life. It was constructed for indexing of different types of Russian texts in a broad domain of sociopolitics (such as official documents or news reports).

The Thesaurus was created in semi-automatic mode using automatic processing of more than 150 Mb of Russian official texts (Lukashevich 1995). The thesaurus units represent real text expressions. In this sense Thesaurus is similar to such thesauri as WordNet (Miller et al. 1990) and Roget's thesaurus. Carefully gathered terms form rows of synonyms for concepts (descriptors of Thesaurus). Adjectives and verbs that are derivatives of a descriptor can also be its variants.

Ambiguous terms can be described in two ways in the Thesaurus. An ambiguous term can be a quasi-synonym of two or more descriptors that represent different meanings of this term. For example, (hereinafter we give fragments from the Thesaurus in English translation) term *capital* is described as a synonym to two descriptors *CAPITAL (City)* and

CAPITAL (Finance). If only one meaning of an ambiguous term is represented in the Thesaurus such term is marked with a special sign of ambiguity.

Existing relationships between descriptors in Thesaurus are: broader term (BT) -- narrower term (NT), related term (RT), whole-term (WT) -- part-term (PT). Latter relationship is used for description of physical parts, elements and objects of a concept.

Using these relations we developed our Thesaurus as a thesaurus inheritance system in which more specific concepts inherit information from more general concepts. In our system this means that relationship "related term" is inherited from a descriptor by its narrower descriptors and by its parts. Relationship "part-term" is inherited from a descriptor by its narrower descriptors. Relationships "broader term --narrower term" and "whole-term --part-term" are transitive relationships.

Thus every descriptor of Thesaurus is related to a wide scope of terms. For most descriptors the number of related descriptors is much larger than the number of direct indicated relationships. For example, descriptor *AGRICULTURE* has 26 direct relations with other descriptors, but through the properties of inheritance and transitivity it is related to more than 300 ones (branches of agriculture, agricultural enterprises, domestic animals and plants and so on).

This extended set of related terms in Thesaurus enables us to determine which terms of the text are semantically or thematically related to each other and can support a topic or a subtopic of the text. As an example, a description of the concept "fishing" is represented on Figure 1.

Currently the Thesaurus contains in Russian more than 21 thousand terms and 8 thousand geographic names (15,000 descriptors and about 40,000 relations between descriptors).

4. Construction of Thematic Representation

In this section we describe our technique of conceptual indexing initially used for processing of Russian texts. The technique was adapted to TREC-6 routing task with insignificant changes.

4.1. Identification of Terms in Texts

Text units are compared with the terms of the Thesaurus using morphological representation of the text and terms. If the same fragment of a text corresponds to different descriptors of the Thesaurus, ambiguity of the text unit is indicated.

Texts can include names that coincide with terms of the Thesaurus. A name that corresponds to a term of the Thesaurus but has different spelling (capital letters, quotes) is also marked as an ambiguous term.

After comparison with the Thesaurus the text is represented as a sequence of descriptors. All synonyms of any descriptor are represented by that descriptor and are not differentiated further. For every text descriptor related text descriptors are given. A set of text descriptors together with relations to related text descriptors is called a "thesaurus projection".

4.2. Disambiguation of Terms Using Thesaurus Projection

Descriptors corresponding to different meanings of ambiguous terms also participate in the construction of the thesaurus projection for a text. Using the thesaurus projection a proper meaning of an ambiguous term is chosen.

For every meaning of an ambiguous term the following conditions are checked. If one of the conditions is met, we consider the text to support this meaning of the ambiguous term.

1) A descriptor corresponding to a meaning of the ambiguous term is used in text in unambiguous form. For example, term *financial capital* is an unambiguous term for descriptor *CAPITAL(Finance)* and *capital* is an ambiguous term for this descriptor;

2) A descriptor corresponding to a meaning of the ambiguous term is related to other descriptors in the thesaurus projection. For example, descriptor *PUBLIC ORGANIZATION* is connected by relationship NT with descriptor *POLITICAL PARTY* that corresponds to one of the meanings of ambiguous term *party*.

РЫБОЛОВСТВО	fishing
UF ВЫЛОВ РЫБЫ; ДОБЫЧА РЫБНЫХ РЕСУРСОВ; УЛОВ РЫБЫ; ПОБЫЧА РЫБЫ; ЛОВ РЫБЫ; ПРОМЫСЕЛ РЫБЫ; ПРОМЫСЛОВОЕ РЫБОЛОВСТВО; ПРОМЫСЛОВЫЙ ЛОВ; ПРОМЫШЛЕННОЕ РЫБОЛОВСТВО; РЫБНАЯ ЛОВЛЯ; РЫБНЫЙ ПРОМЫСЕЛ; РЫБОДОБЫВАЮЩИЙ; РЫБОЛОВНЫЙ; РЫБОЛОВЕЦКИЙ; РЫБОЛОВНАЯ ДЕЯТЕЛЬНОСТЬ; РЫБОПРОМЫСЛОВЫЙ; РЫБОПРОМЫСЛОВАЯ ДЕЯТЕЛЬНОСТЬ	
BT ВОДНЫЙ ПРОМЫСЕЛ UF ПРОМЫСЕЛ ВОДНЫХ БИОРЕСУРСОВ	BT fishery
NT МОРСКОЕ РЫБОЛОВСТВО UF ОКЕАНИЧЕСКОЕ РЫБОЛОВСТВО	NT maritime fishery
NT НЕЗАКОННЫЙ ЛОВ РЫБЫ	NT illegal fishing
NT ПРЕСНОВОДНОЕ РЫБОЛОВСТВО UF ПРУДОВОЕ РЫБОЛОВСТВО	NT freshwater fishing
NT ТРАЛОВЫЙ ЛОВ UF ТРАЛОВАЯ ОПЕРАЦИЯ; ТРАЛОВЫЙ ПРОМЫСЕЛ	NT trawl fishing UF trawling
NT ЛЮБИТЕЛЬСКОЕ РЫБОЛОВСТВО UF ЛЮБИТЕЛЬСКАЯ ЛОВЛЯ; ЛЮБИТЕЛЬСКИЙ ЛОВ	NT
PT РЫБАК UF РЫБОЛОВ	PT fisherman
PT РЫБОЛОВНОЕ ПРЕДПРИЯТИЕ UF РЫБКОЛХОЗ; РЫБОДОБЫВАЮЩАЯ ОРГАНИЗАЦИЯ; РЫБОДОБЫВАЮЩЕЕ ПРЕДПРИЯТИЕ; РЫБОЛОВЕЦКАЯ АРТЕЛЬ; РЫБОДОБЫВАЮЩИЙ ТОВАРОПРОИЗВОДИТЕЛЬ; РЫБОЛОВЕЦКИЙ КОЛХОЗ; РЫБОЛОВЕЦКОЕ ПРЕДПРИЯТИЕ; РЫБОЛОВНАЯ ОРГАНИЗАЦИЯ; РЫБОЛОВНОЕ ХОЗЯЙСТВО; РЫБОПРОМЫСЛОВАЯ ОРГАНИЗАЦИЯ	PT commercial fishery enterprise
PT РЫБОЛОВНЫЕ ОРУДИЯ UF ОРУДИЕ ЛОВА; РЫБОЛОВНАЯ СНАСТЬ;РЫБОЛОВНОЕ СНАРЯЖЕНИЕ	PT fishing equipment
PT РЫБОПРОМЫСЛОВАЯ РАЗВЕДКА	PT fish reconnaisance
PT РЫБОПРОМЫСЛОВЫЙ ФЛОТ UF ПРОМЫСЛОВЫЙ ФЛОТ; РЫБНЫЙ ФЛОТ; РЫБФЛОТ; РЫБОЛОВЕЦКИЙ ФЛОТ; РЫБОЛОВНЫЙ ФЛОТ; ТРАЛОВЫЙ ФЛОТ; ФЛОТ РЫБНОЙ ПРОМЫШЛЕННОСТИ	PT fishing fleet
RT РЫБА UF ВИД РЫБ; РЫБНОЕ СЫРЬЕ; РЫБНЫЙ	RT fish
RT РЫБНАЯ ПРОДУКЦИЯ UF МЯСО РЫБЫ; РЫБНАЯ ГАСТРОНОМИЯ; РЫБОТОВАРЫ РЫБНЫЕ ПРОДУКТЫ; РЫБНЫЕ ТОВАРЫ; РЫБОПРОДУКТЫ; РЫБОПРОДУКЦИЯ	RT fish products
RT РЫБНЫЕ РЕСУРСЫ UF РЫБНЫЕ ЗАПАСЫ	RT fish resources

Figure 1. Example of CIR Thesaurus concept description

If the text supports only one meaning of the ambiguous term the corresponding descriptor is chosen. If the text supports more than one meaning of the term we look through descriptors that are the nearest ones to every usage of the ambiguous term and choose the meaning of the descriptor supported by the nearest descriptors.

Only chosen descriptors participate in further processing of the text.

4.3. Construction of Thematic Nodes

We assume that the term that characterizes a topic of the text and therefore can become the thematic center of a thematic node is usually stressed in a text. It can be used in the title or in the beginning of the text or it can have the highest frequency among terms of the topic.

Any term of the Thesaurus (either general or specific one) can become the thematic center of a thematic node. For example, term *mathematics* can become the main term of a topic if the text is devoted to development of mathematics, or term *scientist* can become the main term of a topic if a text is about "brain drain" to foreign countries.

Creation of thematic nodes begins by choosing the thematic centers. First, descriptors mentioned in the title and first sentence of the text gather all related descriptors from the thesaurus projection and become the thematic centers of thematic nodes. Then the most frequent descriptors of the text can become thematic centers. A descriptor included into a thematic node cannot become the thematic center of a new thematic node.

Let us analyze document FBIS-F001-0015 (Figure 2). Some thematic nodes that were constructed during automatic processing of the example text (the right column represents descriptor frequency in the text) are as follows:

Russia (Russian)	10
Far East	1
Curile	1
President of Russia	1
state (country)	6
territorial waters	9
ocean	3
ship	4
island	1
state (country)	6
President of Russia	1
fish	11
fishing	5
fisherman	2
illegal fishing	1
pouching (pouch)	5
illegal activity	1
illegal fishing	1
fisherman	2

Border Troops `Putina' Exercise to Control Poaching

[Text] The border troops "are not saber rattling" in Russian territorial waters in the Far East as the mass media, especially the Japanese mass media, are attempting to portray it. Servicemen have been legally granted the right to utilize all of the tools at their disposal, including weapons, to put a stop to poaching. Russian Border Troops Commander-in-Chief Colonel-General Andrey Nikolayev stated that to an ITAR-TASS correspondent while stressing that his subordinates are conducting a strict policy to put a stop to the illegal activities of foreign boats. He noted that the President of Russia supports the position of the border troops for the full observance of the law in the country's territorial waters.

Recently, we have become accustomed to reports on the entry of Japanese fishing boats into Russian territorial waters to poach fish. According to official data, the number of such violations has increased by a factor of 3.5-4 in 1993, in contrast to 1990. And although the Russian border guards, who are experiencing great difficulties in logistics-technical support due to the well-known economic situation in the country, have been able to observe approximately 140 foreign fishing boats and to fine poachers a sum of more than 21 million rubles and over 100,000 U.S. dollars in 199, so far, their efforts are a drop in the sea. These fines have hardly made up for the damage from more than 7,500 pirate entries into Russia's territorial waters.

.....

(full text size is about 7 Kb).

Figure 2. Fragment of FBIS-F001-0015 document (terms of four thematic nodes are underlined)

4.4. Determination of Status of a Thematic Node

In the previous stage thematic nodes were gathered. Each thematic node includes descriptors of the thesaurus projection that are related to its thematic center. Thematic nodes correspond to topics or subtopics discussed in a text. At this stage it is necessary to evaluate the importance of topics and thematic nodes representing these topics in the text. The first step is to determine main topics of the text, that is to choose main thematic nodes.

In our approach we assume that in normal, conventional texts main topics pass through the whole text and are discussed in combination with each other. This means that descriptors of different main thematic nodes are usually located together all over the text. To find out how descriptors of thematic nodes are distributed in the text we use the notion "textual relation": a given descriptor has textual relations with those descriptors of the text that are located not further than N descriptors from the given descriptor (location order is not important). Currently N=2, so every usage of a descriptor in the text is considered in a sequence of descriptors by length 7. Thus we assume that in a text descriptors of thematic nodes are usually repeated over seven descriptors. This approach originates on the basis of experiments in psychology and linguistics.

As a result we obtain a set of textual relations for every descriptor of a text.

Textual relations between descriptors are determined at the stage of comparison of text with Thesaurus. After thematic nodes are constructed, textual relations frequencies of descriptors in each thematic node are summed to compute the textual relations between thematic nodes.

In our approach we assume that first of all main thematic nodes are those ones that

- have textual relations with all other main thematic nodes and
- have a sum of frequencies of textual relations between these nodes greater than the sum of frequencies for the same number of other thematic nodes of this text.

The thematic nodes for the example in Figure 1 are thematic nodes with main descriptors *territorial waters, fish, Russia, Japan, border troops, poaching, boat, ...*
Thus we can produce a "thematic summarization" of text (right column represents total frequency of thematic node descriptors):

territorial waters; state (country);ship; ocean; island; President of Russia	24
fish; fishing; fisherman; illegal fishing;	20
Russia (Russian); state (country); Far East; Curile; President of Russia	19
Japan; continental shelf; state(country)	18
border troops; border guard, state(country)	17
pouching (pouch); fisherman; illegal activity; illegal fishing	9
boat	7

These requirements for main thematic nodes determine a threshold that distinguishes main thematic nodes form all other thematic nodes of a text. This threshold is an average frequency of descriptors in determined main thematic nodes. The initial set of main thematic nodes is supplemented with those thematic nodes whose frequency is more than the threshold.

In addition to main thematic nodes there are specific thematic nodes and mentioned descriptors. Specific thematic nodes represent primary characteristics of main topics discussed in the text. Specific nodes are those thematic nodes that have textual relations with at least two different main thematic nodes. Descriptors that are not elements of main or specific thematic nodes are called mentioned descriptors.

In our example specific thematic nodes are:

logistics
equipment
monitor
computer

mass media
correspondent

The first one is represented in the following paragraphs of example (Figure 3). Mentioned descriptor are *weapon, expert, ice situation ....*

..... (3rd paragraph)
But then again, we can explain the definite impunity of violators not only through the problems in logistics-technical support, due to which border troops maritime units and aircraft have been compelled to reduce their activities (for example, last year the United States had 3.2 ships per 100,000 square kilometers of economic zone, Japan had 8.2, and Russia had 2.1), but also through the obvious delay in the adoption of the laws "On the Russian Federation's Exclusive Economic Zone" and "On the Russian Federation's Continental Shelf",..
..... (8th paragraph)
It is noteworthy that the poachers' schooners have been well adapted for "wolf-like" swoops into our territorial waters. They have excellent navigation equipment, they are equipped with computers and they are maneuverable. Maneuverability also helps them to feel quite confident in themselves even under conditions of a complex ice situation (up to 4-5 balls)....

Figure 3. Fragments of FBIS-F001-0015 document

Thus all descriptors of the text are divided into five classes of decreasing importance for the text:
- main descriptors of main thematic nodes,
- other descriptors of main thematic nodes,
- main descriptors of specific thematic nodes,
- other descriptors of specific thematic nodes,
- mentioned descriptors.

5. Text Categorization Using Thematic Representation of Text

The thematic representations of texts can serve as a basis for text categorization. It was used for processing of TREC-6 routing task when TREC-6 topics were described as categories for text categorization.

5.1. Relations between the Thesaurus and Categories

Our technique allows us to carry out text categorization using different systems of categories.
We consider a category to be a user defined query that has to be represented by descriptors of the Thesaurus. The hierarchical structure of the Thesaurus allows to choose a subtree of the Thesaurus corresponding to the category and connect the category with upper descriptor of this subtree. We call such a descriptor "supporting descriptor" of the category.

A category can be represented by a set of descriptors. We define two types of category representation over a set of supporting descriptors.

The first type of representation is a disjunction of supporting descriptors

$$D_1 \cup D_2 \cup \ldots \cup D_n. \quad (1)$$

For example, the category "Taxes and Budget" can be represented with expression *TAX* $\cup$ *BUDGET SYSTEM*.

The other type of representation is a conjunction of disjunctions of supporting descriptors

$$(D_{11} \cup D_{12} \cup \ldots \cup D_{1n}) \,\&\, \ldots \,\&\, (D_{21} \cup D_{22} \cup \ldots \cup D_{2m}) \,\&\, \ldots \,\&\, (D_{k1} \cup D_{k2} \cup \ldots \cup D_{kr}). \quad (2)$$

For example, category "Taxes and Budget of the Russian Federation" is represented with the following sequence of supporting descriptors: *(TAX ⋃ BUDGET SYSTEM) & RUSSIAN FEDERATION.*

After relations between categories and supporting descriptors are fixed, categories corresponding to other descriptors of the Thesaurus are established automatically using the hierarchy of Thesaurus. As a result most descriptors of the Thesaurus are connected with some categories indicating the disjuncts it belongs to. A descriptor can have no category.

5.2. Text Categorization Using Different Systems of Categories

Text categorization of official documents of the Russian Federation is fulfilled for Information System RUSSIA (Yudina & Dorsey 1995). The system of categories consists of 180 categories that are connected with 210 supporting descriptors of the Thesaurus. Categories are represented as disjunctions of supporting descriptors. (Loukachvitch, 1997).

Text categorization for news reports uses 35 categories that are connected with 145 supporting descriptors of the Thesaurus. Most categories are represented as conjunctions of two disjunctions of supporting descriptors.

To provide convenient access to Russian official documents via the Internet for users accustomed to one of well-known thesauri (LIV 1990; UNBIS THESAURUS 1976), we took top categories (top terms, subject headings) from these thesauri and created relations between the categories and our Thesaurus. Every such thesaurus has a systematic part describing correspondence between its descriptors and top categories. Thus these systematic parts determine interpretation of each top category. For example, Legislative Indexing Vocabulary (LIV 1990) has 89 top terms that were connected with 250 supporting descriptors of our Thesaurus. In particular, top term "Medicine" containing 400 descriptors in LIV was connected with 7 supporting descriptors and currently 460 descriptors of our Thesaurus correspond to this top term.

6. TREC-6 Routing Task

We assumed that after matching a text with thesaurus units the remainder of our technique is language-independent. Thus to process TREC text collections we had to perform the following tasks:

- supplement the English translations of Thesaurus terms with synonymic expressions (size of Russian synonymic rows reach 20 and more elements);
- create morphological analyzer of English words;
- describe ambiguity of English terms by means of our Thesaurus;
- represent topics of TREC-6 as logical expressions of supporting descriptors.

6.1. Description of TREC-6 Topics

TREC-6 routing task carried out by Center for Information Research was close to the general strategy of CIR for automated text processing.

We used manually query construction where TREC-6 topics were represented as categories for text categorization.

Each topic was described as logical expression:

$$\bigcup X_i = \bigcup (\& x_{ij}) \ .$$

For each operand x_{ij} some supporting descriptors from the Thesaurus were chosen. After that the query was expanded by narrowed descriptors from Thesaurus.

Finally

$$x_{ij} = \bigcup w_{ijk} \ ,$$

where w_{ijk} descriptors from Thesaurus.

For example the query for Topic 012 "Water Pollution - document is about the pollution of a body of water" was defined as:

$$X_1 \cup X_2$$

$$X_1 = x_{11};\ x_{11} = \text{A};\quad X_2 = (x_{21} \,\&\, x_{22});\ x_{21} = \text{B};\ x_{22} = \text{C}$$

Figure 4 gives the detailed description of TREC-6 topic 012.

	x_{ij}	w_{ijk}
012	A "water pollution"	federal water pollution control act; federal water pollution control administration; hot water pollution; sewage disposal pollution of sea environment; sewage water pollution; water purification water supply and pollution control division
012	B "pollution"	ground pollution; oil distribution supertanker shipwreck oil pollution; oil spill
012	C "body of water"	body of water; animalis aquaticus; basin; fresh water; freshwater fishing; freshwater aquaculture; inland waterways freshwater reservoir; maritime fishery; lake ocean; ocean resources; reservoir; river; salt water; sea; sea animal; sea fish sea flora; sea mammal; sea-water; water basin sources of water; surface waters; water biological resources; water plant water resources; water scoop; water supply water-way; watershed

Figure 4. Topic 012 description

6.2. Processing Documents

We created an English morphological analyzer using standard morphological rules and WordNet exception lists. A morphological representation was built for every English entry of the Thesaurus.

During processing of a document we calculated the weights of any topics that were found.

The general rule was

$$\mu_D = \max_i(\ \mu_X(X_i)\ )\ ,$$

where weight of operand group is:

$$\mu_X(X_i) = \Pi_j\ \mu_x(x_{ij}) = \mu_x(x_{i1}) \cdot \mu_x(x_{i2}) \cdot \ldots \cdot \mu_x(x_{im})\ ,$$

weight of operand calculated as:

$$\mu_a(x_{ij}) = max\{\mu_0 , \nu_T(w_{ijk})\} ,$$

here $\mu_0 = 0.001$,

$$\nu_T(a_{ij}) = \begin{cases} 1.00, & \text{if } a_{ij} \text{ represents the main descriptor of main thematic node,} \\ 0.60, & \text{if } a_{ij} \text{ represents a descriptor of main thematic node,} \\ 0.30, & \text{if } a_{ij} \text{ represents the main descriptor of specific thematic node,} \\ 0.10, & \text{if } a_{ij} \text{ represents a descriptor of specific thematic node,} \\ 0.05, & \text{if } a_{ij} \text{ represents a mentioned descriptor,} \\ 0.00, & \text{otherwise.} \end{cases}$$

7. Analysis of Results

Our TREC6- routing results are close to median of the Category A routing results thus confirming the basic principles of our technology.

During our TREC-6 processing we encountered the following problems:

- ambiguity of English terms considerably differs from ambiguity in Russian and its description requires additional information;
- some subunits of TREC-6 topics could not be expressed by means of our Thesaurus.

The Thesaurus is to be further developed and carefully horned and tested in order to obtain better results using our technology of conceptual indexing for English texts.

Bibliography

Brazilay R., Elhadad M. 1997. Using Lexical Chains for Text Summarization. - ACL/EACL Workshop Intelligent Scalable Text Summarization.- Madrid.

Climent S., Rodriguez H., Gonzalo J. Definitions of the links and subsets for nouns of the EuroWordNet project. - Deliverable D005, WP3.1, EuruWordNet, LE2-4003.

van Dijk T.A., Kintsch W. 1983. Strategies of Discourse Comprehension. New York. Academic Press, 1983.

Halliday M., Hasan R. 1976. Cohesion in English. Logman, London.

Hirst G., St-Onge D. 1997. Lexical Chains as representation of context for the detection and correction malapropisms. In C. Fellbaum, editor, WordNet: An electronic lexical database and some of its applications. Cambrige, MA: The MIT Press.

LIV 1990. Legislative Indexing Vocabulary 19th Edition. - Washington: The Library of Congress.

Loukachevitch N. 1997. Knowledge Representation for Multilingual Text Categorization . AAAI Symposium on Cross-Language Text and Speech Retrieval, AAAI Technical Report, 1997, pp. 133–142.

Lukashevich N. 1995. Automated Formation of an Information-Retrieval Thesaurus on the Contemporary Sociopolitical Life of Russia. *Automatic documentation and mathematical linguistics*. 29(2): 29-35.

Miller G., Beckwith R., Fellbaum C., Gross D. and Miller K. 1990. Five papers on WordNet. CSL Report 43. Cognitive Science Laboratory, Princeton University.

Salton G. 1989. Automatic Text Processing - The Analysis, Transformation and Retrieval of Information by Computer. Addison-Wesley, Reading, MA.

UNBIS Thesaurus 1976. English Edition.- Dag Hammarskjold Library of United Nations, New York.

Subject Headings 1991. Subject Headings. 14th Edition. - Cataloging Distribution Service, Library of Congress, Washington, D.C.

Yudina T., Dorsey P. 1995. IS RUSSIA: An Artificial Intelligence-Based Document Retrieval System. *Oracle Select.* 2(2), 12-17.

Experiments in Query Optimization

The CLARIT System TREC-6 Report

Natasa Milic-Frayling, Chengxiang Zhai, Xiang Tong, Peter Jansen, David A. Evans

CLARITECH Corporation
A Justsystem Group Company

1 Introduction

The CLARITECH team completed five TREC-6 tasks: the two traditional TREC tasks, Routing and Ad-Hoc Retrieval, and three special tracks, Filtering, Chinese, and Spoken Document Retrieval. We performed TREC-6 experiments in the newly developed CLARIT System Evaluation Environment that is based on CLARIT System APIs, developed at CLARITECH Corporation.

In general, all CLARIT processing for TREC-6 tasks (except Chinese) took advantage of standard CLARIT indexing, which involves a natural-language processing of source texts to identify and normalize noun phrases, sub-phrases, and individual words. In addition, most processing involved one or more methods for the identification of terms to supplement a query or information profile, including the traditional CLARIT method of Thesaurus Discovery, which automatically identifies salient terminology in document texts. However, we also applied newly implemented CLARIT System features for (1) profile training and query expansion through multi-pass training and feedback processes and (2) automatic learning of the optimal system configuration for a particular topic or a particular evaluation criterion.

In previous TREC experiments, the CLARITECH team focused on the optimization of system performance at a "macro" level, targeting the whole set of TREC topics, maximizing the effects of various approaches without further refinement at the level of individual topics. While our efforts resulted in high performance, it was clear that there were many effects at the level of individual queries that were not well understood or optimized. Thus, our work has naturally evolved toward a focus on the use of multiple methods to capture various aspects of a user's information needs and toward techniques that can automatically optimize a user-defined utility on a per-topic basis. Much of our work in TREC 6 reflects such concerns.

We have exploited three principal mechanisms to make our processing more sensitive to the requirements of individual queries: (1) term-selection methods (as applied to profile training and query expansion); (2) combinations of evidence to establish document relevance (as applied in performance optimization for individual topics in document routing); and (3) calibration of relevance scores and relevance thresholds on a topic-by-topic basis (as applied in document filtering). Such mechanisms are reflected in several of our experiments, especially in the routing and filtering tracks.

In the following sections we describe the CLARIT TREC-6 system configuration and present the results of our experiments in each track. There are many details to the processes that we employed and many complexities in our approach to specific tasks. (We conducted more than 1,000 preliminary experiments to examine the behavior of some of the parameters of the system. The observations we made based on such experiments were distilled in the design of our final system configurations.) We have made an effort to provide the reader with sufficient details to follow our discussions while, necessarily, omitting a great deal of background information.

We emphasize that the CLARIT system is highly parameterized. Consequently, the exploration of the parameter space is involved and challenging. For the purpose of this presentation, we deliberately restrict our discussions to the main issues illustrated in our official experiments. For details on other system features we refer the reader to our reports from previous TRECs. (See [Evans et al. 1993, 1996]; [Evans & Lefferts 1994, 1995]; [Milic-Frayling et al. 1996, 1997]; [Zhai et al. 1997]; [Tong et al. 1997a,b].)

2 The CLARIT System Evaluation Environment

The CLARIT System is a suite of information management tools, all based on a common software architecture that has been implemented in C++ through a highly object-oriented design and with special attention to re-usability and extensibility of code. We use the commercial CLARIT APIs in our research and prototype development work. Specifically, for the CLARIT TREC-6 evaluations, we configured CLARIT modules to support elaborate and flexible experimentation with various techniques in ad-hoc retrieval, document routing, and document filtering. (See Figures 1 and 2 for a schematic representation of the system configuration used in our routing and ad-hoc experiments.) Here we briefly describe the components of the CLARIT System that were used in our experiments.

2.1 CLARIT NLP

The CLARIT NLP Module consists of a parser and a morphological analyzer that use an English lexicon and grammar to identify linguistic structures in text. It supports the discovery of various types of linguistic structures, including simplex and complex noun phrases (NPs), verb phrases, prepositional phrases, and selected other constituents. In all our TREC-6 experiments, except for the Chinese Track, we used simplex noun-phrases as a basis for text processing. (See [Evans 1990] and [Evans et al. 1991] for early descriptions of CLARIT NLP design and [Evans & Zhai 1996] for more recent work.)

2.2 CLARIT Indexing

CLARIT Indexing involves statistical analysis of a text corpus and construction of an inverted index that allows the system to retrieve documents or parts of documents that contain a given set of terms. Typically, we invert the index for full documents or for each variable-length subdocument in the document. The length of subdocuments is not strictly uniform but rather specified as range of lengths around a target average number of sentences that should be included in a subdocument. More precisely, when any paragraph boundary falls within the specified range, the actual length of the subdocument is determined by that boundary; otherwise, it is equal to the specified average number of sentences.

The indexing module supports various levels of term space granularity, including noun-phrases, their single-word constituents, attested subphrases, etc. In all our TREC-6 experiments, except for the Chinese track, we use database indices that consist of noun-phrases and single-word constituents. In the case of Ad Hoc processing, we used attested sub-phrases as well.

2.3 CLARIT Retrieval

Document Relevance Measures

The CLARIT System supports various document relevance measures that naturally arise from the vector space retrieval model. In TREC-6 experiments we used the dot product similarity function

$$Score(P,D) = \sum_{t \in P \cap D} W_P(t) \cdot W_D(t)$$

where $W_P(t)$ is the weight associated with the query or profile term

$$W_P(t) = C(t) \cdot TF_P(t) \cdot IDF_P(t)$$

and $W_D(t)$ is the weight associated with the term in the document

$$W_D(t) = TF_D(t) \cdot IDF_D(t).$$

In the above formulas, IDF and TF are standard inverse document frequency and term frequency statistics, respectively. Typically, $IDF_D(t)$ and $IDF_P(t)$ refer to the same statistics: either IDF in the reference corpus for routing applications or IDF in the target corpus for retrieval applications. The coefficient C(t) is an "importance coefficient" and can be determined either manually by the user or automatically by the system.

The CLARIT evaluation system allows flexible setting of TF and IDF statistics. In particular, TF or IDF factors can be set to a constant value to achieve different term weighting effects. For example, if both TF and IDF are set to 1 for all the terms in the profile and the document, the resulting relevance score becomes the sum of term importance coefficients. If, in addition, we set all importance coefficients to 1, the relevance score represents the count of overlapping terms between the document and the profile. Thus, a variety of similarity measures, ranging from more traditional vector-space scoring to weighted or simple feature counts, can be used as required in different processing circumstances.

Document Working Set

In order to speed up relevance scoring of documents in the database, the CLARIT System uses a document working set feature that allows the system to score only the documents that are most likely to be relevant to a particular profile or query. The working set represents a subset of the total set of documents that will be considered for relevance scoring. The terms in the profile or the query are ranked in decreasing order of distribution statistics. Documents that contain the least distributed term are added first to the working document set. Then the second-least distributed term on the list is considered and all the documents that contain that term and that are not already included in the working set are added. The procedure continues until all the terms in the profile or the query are exhausted or the working set reaches the maximum size specified by the user.

This approach to determining a working set is based on the assumption that terms with lower distribution (i.e., higher IDF scores) have a higher discriminating power in distinguishing relevant documents from non-relevant ones, and that, ultimately, the highest scoring documents will be those that contain such terms.

Subdocument Matching

Documents can be processed by the system as full documents or as collections of subdocuments. The primary reason for using subdocuments in profile training and updating is to obtain segments of relevant documents that contain concentrated relevant terminology. During document/profile and document/query relevance matching, subdocuments are potentially useful for normalizing the generalized feature matching score and reducing possible bias towards longer documents.

In our TREC experiments we use the score of the best scoring subdocument to score and rank a document. Such a document scoring technique is commensurate with the specific relevance model adopted by TREC: a document is judged relevant if any of its parts, possibly as short as a single sentence, is relevant to the topic.

Constraints

The CLARIT System supports natural language queries supplemented by Boolean type constraints. Constraints are used as filters on the retrieved set of documents. They provide an effective means for enforcing the user's relevance criteria. In our TREC-6 experiments, we use constraints in the initial formulation of ad-hoc queries to support the selection of subdocuments for automatic query expansion.

2.4 Terminology Discovery Module

Many tasks in information management, such as automatic query augmentation and profile training, involve a process of selecting salient terminology from a set of documents. Typically, such terminology is selected to characterize a given document set in contrast to some general collection of documents. Following this approach, the CLARIT term selection methods take as input a document set to be characterized and a background database to provide contrasting term statistics (e.g., term distribution statistics or IDF scores). The output is a ranked list of all the terms (phrases or words) found in the specified documents. By applying a cut-off to this list, we can select a portion of the most prominent terms to be added to the query or the profile.

There are various ways in which term selection methods can be integrated into information processing. In retrieval situations with relevance feedback, we apply these methods to documents judged relevant and those judged non-relevant by the user, to extract "positive" and "negative" terminology, respectively. The extracted terminology is added to the query to refine the search. A similar procedure is used for profile generation and refinement in routing applications when "positive" and "negative" examples for a given topic are available. Furthermore, information obtained from the term selection process can be propagated through other processes in the system. For example, some term selection methods involve term weighting that can be used to provide a finer differentiation among profile or query terminology and improve relevance matching (e.g., in probabilistic term weighting).

In our TREC-6 experiments, unless otherwise specified, we used only positive examples (i.e., documents that a user judged as relevant), and not negative ones, for term selection. Furthermore,

since the CLARIT system uses phrase indexing, all term selection methods are applied to the term space consisting of linguistic phrases and single words.

Here we describe the three approaches to term selection that we explored in our official TREC-6 routing, filtering, and ad-hoc experiments.

CLARIT Thesaurus Discovery

The CLARIT Thesaurus Discovery (CLThes) involves a proprietary process for combining multiple statistics on terms in an identified set of documents. All the terms in selected documents are ranked with respect to these statistics; terms that rank consistently high on all criteria are included in the "first-order" thesaurus. The top N% of the terms is used for profile generation or query expansion, where N is empirically determined for a particular application.

CLARIT Probabilistic Term Relevance Measure

In situations when information about relevant documents in a given corpus (e.g., a training corpus or some type of reference corpus) is complete and reliable, it is appropriate to use a probabilistic term weighting to rank and select terminology for query or profile refinement. To explore such effects, we implemented the standard Robertson-Sparck Jones formula (see e.g., [Robertson & Sparck-Jones 1976]):

Standard Probabilistic Term Relevance Measure (Prob):

$$Prob(t) = \ln\left(\frac{N-R}{N_t-R_t}-1\right)-\ln\left(\frac{R}{R_t}-1\right)$$

where

N = the number of documents in a reference corpus,
N_t = the number of documents in the reference corpus containing term t,
R = the number of relevant documents, and
R_t = the number of relevant documents containing term t.

However, in practice, the given set of relevant or non-relevant documents typically represents only a subset of such documents and therefore the formula needs to be modified to compensate for incomplete information. In our attempt to address this problem, we modified the formula to increase the influence of term distribution in known relevant documents.

CLARIT Probabilistic Term Relevance Measure (CLProb):

$$CLProb(t) = \ln(R_t+1)\bullet\left[\ln\left(\frac{N-R}{N_t-R_t}-1\right)-\ln\left(\frac{R}{R_t}-1\right)\right]$$

Our modification of the standard probability formula is motivated purely by empirical observations and does not follow from a formal probabilistic model. However, the results are very encouraging and we intend to continue our research in this direction.

For the TREC-6 experiments we implemented a version of the formula that bypasses the problem of singular values without applying any formal smoothing methods. We anticipate a need for a more

rigorous approach to smoothing when we decide to incorporate term scores into relevance matching functions.

The Rocchio Formula

In some experiments, we used the standard Rocchio formula to rank terms in a given set of documents. More precisely, we used term distribution statistics (IDF) from a reference corpus to provide a TF-IDF weighting of terms in the documents and then applied the Rocchio formula to compute the centroid vector for the given set of documents. The coordinates of the centroid vector are taken as term weights and used to rank and select terms:

$$TFW(t) = IDF(t) \cdot \frac{\sum_{D \in DocSet} TF_D(t)}{NumDoc}$$

where

IDF(t)	= Inverse Document Frequency of term *t* in reference database
NumDoc	= Number of documents in the given set of documents
$TF_D(t)$	= Term frequency score for term *t* in document D.

3 CLARIT Routing

The main issue explored in our TREC-6 Routing experiments is the effectiveness of multiple term-selection techniques to create a profile or a set of profiles for a topic. We evaluated two techniques for exploiting multiple term-selection methods: (1) combining term-selection methods to optimize the routing performance for a set of profiles and (2) identifying, for each individual profile, a term-selection method that optimizes routing performance for that profile.

More specifically, we designed a two-pass profile refinement process that supports different term selection methods in each pass. Applying various combinations of term-selection methods enabled us to generate several versions of profiles for each topic. Among those, we selected the version that performed best over the training data. This was our first attempt towards calibration of the system to maximize the performance for individual topics.

For official submission we selected two runs, CLCOMB and CLMAX, that illustrate the two principal strategies: multi-pass profile training and per-topic optimization of the system. We describe these two approaches in more detail in Sections 3.3 and 3.4, respectively. In the following two sections, we describe our general experimental design and the observations we made in a variety of training experiments.

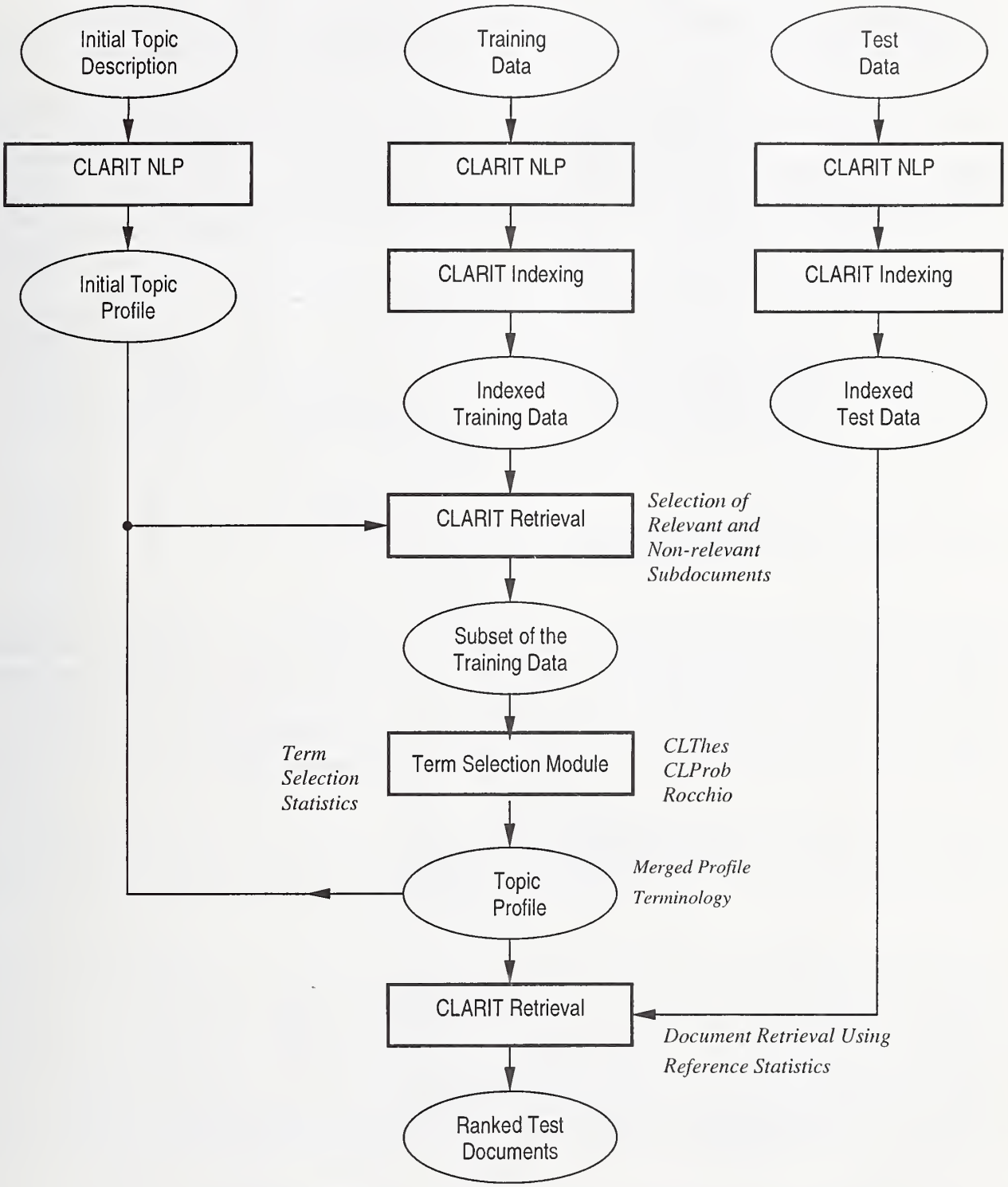


Figure 1. Configuration of the CLARIT Modules in the TREC-6 Routing Experiments.

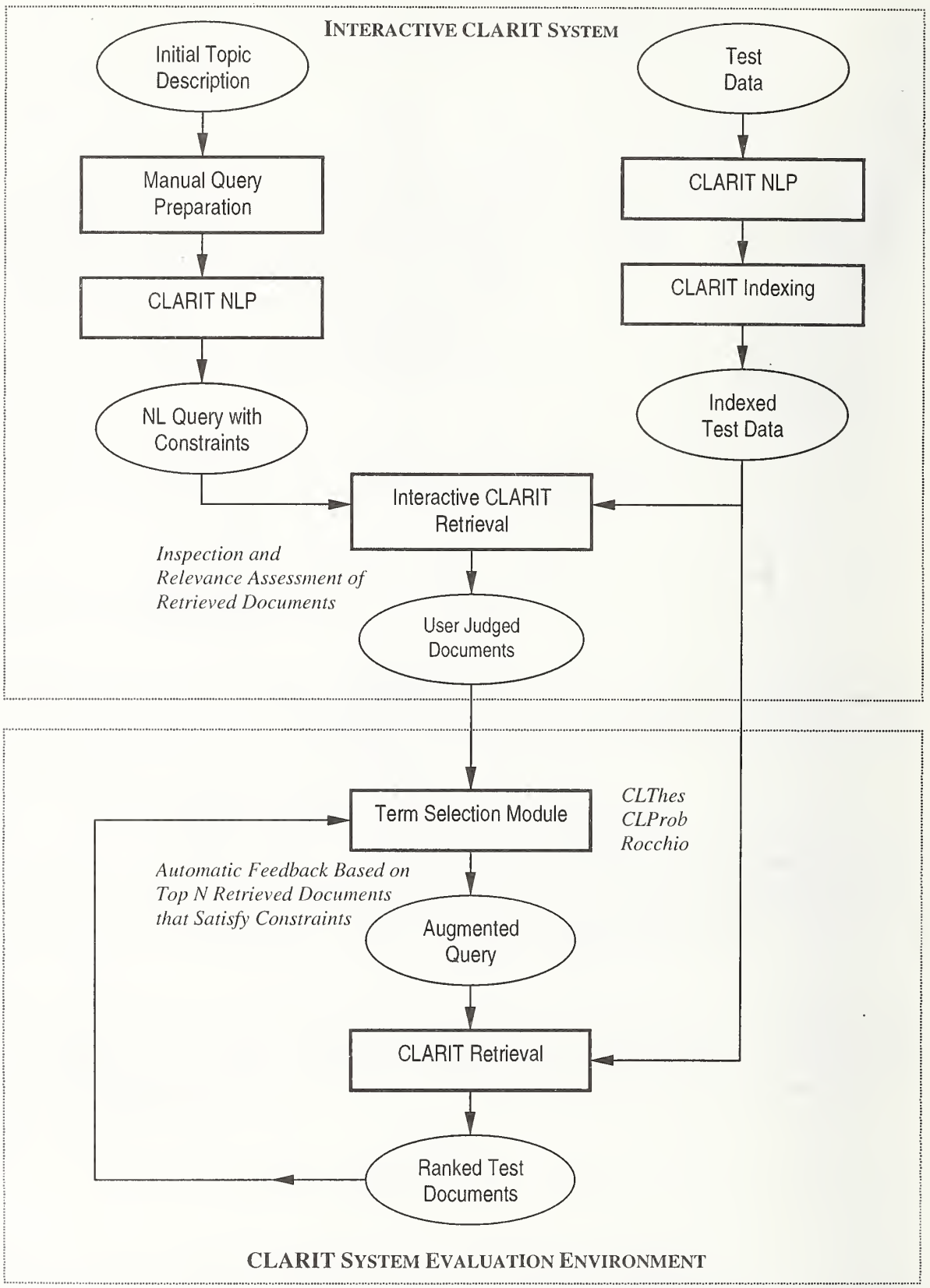


Figure 2. Configuration of CLARIT Modules in the TREC-6 Ad-Hoc Experiments.

3.1 Experiment Design

Our preliminary experiments with the CLARIT System suggested a routing procedure that involves a three-step profile building and refinement process:

1. *Identify the "best" subdocuments from the set of relevant training examples.*
   We achieve this by executing the topic description or an initial profile as a query over a database of training documents. For a given topic, we select subdocuments that are ranked among the top 8,000 retrieved subdocuments and belong to documents judged relevant for that topic. By doing this we essentially exclude non-relevant portions of training documents for a topic.

2. *Apply one of the CLARIT term selection methods to identify profile terminology from the selected subdocuments.*
   Terms in the selected subdocuments are ranked by a given term weighting algorithm. The background statistics used by the term selection algorithms are typically collected over selected subdocuments or some other set of training data. The N top ranked terms are considered for profile generation or expansion.

3. *Add the selected terms to the initial profile vector.*
   If the term is already present in the profile vector, its existing importance coefficient is increased by 0.5. Otherwise, the term is assigned an importance coefficient of 0.5.

In general, we restrict profile training to a subset of the training data, e.g., only those training documents or portions of training documents that respond to the initial topic formulation or initial profile. In addition to the features that carry "positive information", the system can identify "distracting" terminology from non-relevant portions of relevant documents and retrieved portions of non-relevant documents. We used this technique in previous TRECs (see [Evans et al. 1994, 1996]) but in TREC-6 we focused primarily on the evaluation of methods for extracting positive features.

We note that the Cornell group has begun using similar techniques for profile training ("query zone") in recent TRECs [Buckley et al. 1997]. This was preceded by the work of the Xerox group that explored a concept of the "local region" (see [Schutze et al. 1995]).

In our TREC-6 experiments we simulated the routing of documents by using vector-space retrieval, modified to use distribution statistics (IDF) from a reference corpus instead of the target test corpus. The ranks and scores determined by term selection methods are expected to reflect the relative importance of terms in the profile. However, as already observed and explored by other TREC participants (see [Buckley & Salton 1995]) and as we experienced in our preliminary experiments, optimal profile terminology weighting does not follow directly from the term selection process. Furthermore, it remains a challenge to incorporate term weights appropriately into the relevance matching function in vector-space retrieval.

The CLARIT evaluation system does support the separation of profile term selection and profile term weighting. This allows us to study not only the interaction between term selection and term weighting, but also a wide variety of document routing problems, including situations in which training data is scarce or not available. In such circumstances, profiles can be generated manually by the user; and a reference source of term statistics, instead of natural training statistics, can be used to differentiate among profile terms. In our TREC-6 experiments, we made no attempt to use

term weights from the term selection algorithms as a basis for relevance matching of profiles and documents. Instead, profile terms were weighted at the time of document routing based on term importance coefficients and term IDF scores in a reference corpus, i.e., the statistics used for relevance matching.

3.2 Training Data

For the training of TREC-6 profiles, we had the TREC-5-FBIS data and an extended version of FBIS data ("ExtendedFBIS") that includes, in addition to TREC-5-FBIS data, all other training documents that were judged for at least one of the routing topics. Since the TREC-6 routing test data includes only FBIS documents, the TREC-5-FBIS database represents a natural set of training examples. On the other hand, the ExtendedFBIS database is artificially populated with relevant documents from other databases. It does provide a richer set of examples and, therefore, a better source of profile terminology. However, term statistics from that database are likely to differ significantly from term statistics in the test data.

Training data is used in CLARIT Routing as a source of (1) profile terminology; (2) background statistics for ranking and selecting profile terminology; and (3) reference statistics for relevance matching of profiles and test documents (profile term weighting). For different experiments we used different sources or combinations of sources of training statistics, as illustrated in Figure 3.

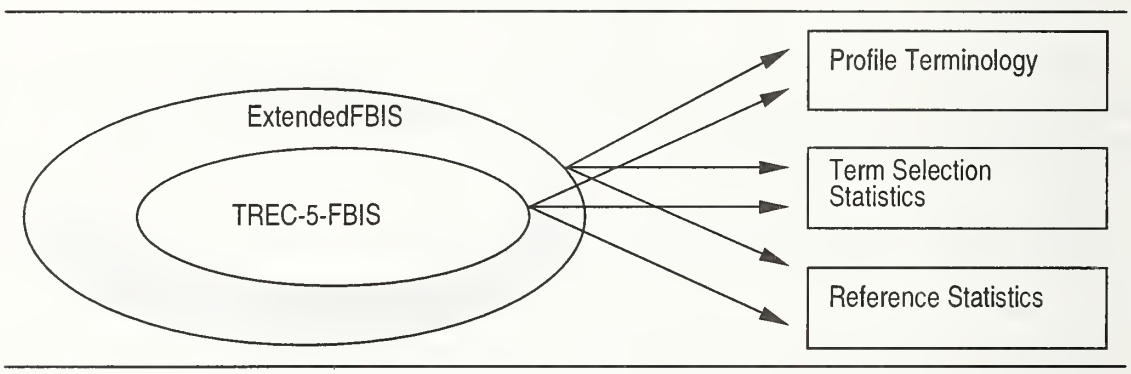


Figure 3. Sources of terminology, term selection statistics, and reference statistics in the CLARIT TREC-6 experiments.

Each profile gets its source terminology from selected subdocuments from the relevant documents for the profile in the training data (either TREC-5-FBIS or ExtendedFBIS). Background statistics used for term ranking may come from the set of selected subdocuments or a completely different set of training data. For example, in some of our experiments we used the terminology found in selected subdocuments from the ExtendedFBIS database as candidate profile terms. The relevance ranking and final selection of these terms, on the other hand, is based on statistics from the TREC-5-FBIS database.

Since sources of term statistics serve essentially as *contrast sets* for the collection of the subdocuments nominating potential profile terms, they provide a context for determining the degree of specificity of terms for the topic. Combining various terminology sources and statistics sources allows us to tune profiles from the perspective of different domains.

Similarly, sets of reference statistics, or, more precisely, term IDF statistics used for profile term weighting in vector space relevance matching, are based on various collections of training data. Since we currently make no attempt to score terms in the final profile except for the assignment of slightly variable importance coefficients, IDF statistics serve to differentiate among terms. Thus, the statistical characteristics of the reference database essentially determine the relative importance of terms in the profile.

3.2.1 Effects of Training Data

In order to understand the subtle interactions among sources of terminology and sources of statistics for terminology weighting and relevance matching, we conducted a number of preliminary experiments. For illustration, we show in Table 1 the different sources of terminology and statistics that were combined in some of our experiments.

Training Examples (Sources of Relevant Documents)	Term Selection Statistics	Reference Statistics (term weighting)
TREC-5-FBIS	TREC-5-FBIS	TREC-5-FBIS
ExtendedFBIS	ExtendedFBIS	ExtendedFBIS
	TREC-6-FBIS	TREC-5-FBIS – part 1
		TREC-5-FBIS – part 2
		TREC-6-FBIS
		AP89

Table 1. Combinations of the training example sets, term selection statistics, and reference statistics used in the CLARIT TREC-6 experiments.

Our experiments revealed that (1) using relevant examples from a richer set of training data generally yields more effective profiles and (2) routing performance is significantly affected by the selection of reference statistics.

Indeed, in Table 2, we see that using training examples selected from the richer set of training data, the ExtendedFBIS database, yields consistently better performance for a fixed set of term selection and reference statistics.

Table 3 shows clearly that, for the same set of profiles (viz., CLProb profiles constructed from examples in the ExtendedFBIS database), the selection of reference statistics for similarity matching and, essentially, for profile term weighting, has a significant impact on the routing performance. It is interesting, but not unexpected, that the IDF statistics from the test database, TREC-6 FBIS, do not necessarily result in the best profile term weighting for the purpose of relevance matching. In fact, we see from Table 3 that some subsets of the TREC-5-FBIS data serve as better sources of reference statistics.

It is clear that, in our approach, the choice of reference statistics is very important for appropriate topic modeling. We generally observe that the weighting of terms in a specific profile is more effective when the reference data are similar to the target data.

Source of Training Examples	Term Selection and Reference Statistics	Recall	InitPr	AvgPr	PrAt10Docs	ExactPr
ExtendedFBIS	TREC-5-FBIS	**4877**	**0.8016**	**0.2780**	**0.4872**	**0.3305**
	ExtendedFBIS	**4867**	**0.7957**	**0.2890**	**0.5043**	**0.3346**
TREC-5-FBIS	TREC-5-FBIS	4696	0.7460	0.2658	0.4830	0.3020
	ExtendedFBIS	4589	0.6635	0.2504	0.4596	0.2949

Table 2. Effect of various sets of term selection statistics. The same statistics are used for relevance matching. Tested on the TREC-6-FBIS data.

Topic Profiles	Reference Statistics	Recall	InitPr	AvgPr	PrAt10Docs	ExactPr
CLProb Generated from examples in ExtendedFBIS	TREC-5-FBIS part2	**4814**	**0.7620**	**0.2775**	**0.5106**	**0.3173**
	TREC-5-FBIS part1	4754	0.7649	0.2739	0.4957	0.3203
	TREC-6-FBIS – *Test Data*	**4778**	**0.7399**	**0.2705**	**0.4936**	**0.3147**
	TREC5-FBIS	4758	0.7359	0.2684	0.5106	0.3153
	ExtendedFBIS	4589	0.6635	0.2504	0.4596	0.2949
	AP89	4557	0.6911	0.2391	0.4681	0.2804

Table 3. Effect of various sets of reference statistics. CLProb profiles generated over the ExtendedFBIS database. Tested on the TREC-6-FBIS data.

3.3 Term Selection Methods

In our preliminary routing experiments we tried a half-dozen different term selection methods for profile building. For our TREC-6 experiments, we selected three representative methods: CLARIT Thesaurus Extraction (CLThes), CLARIT Probabilistic Weighting of terms (CLProb), and standard Rocchio term weighting (see Section 2.4). In part, our choice was designed to maximize the differences among approaches. More precisely, we chose methods that apply significantly different criteria in selecting features from the text, with the intention of determining whether combinations of such methods would yield more effective profiles.

Indeed, our testing of individual methods revealed rather different performance patterns. Some of the methods typically yield high initial precision but low average precision and recall, while others achieve high average precision, but significantly lower initial precision. Manual inspection of the profiles revealed that the terms selected by different methods can differ significantly. This motivated us to explore the effects of combining term selection methods, which led to our first official submission, CLCOMB.

Based on our preliminary experiments, we concluded that a simple merging of terms selected by different methods, with no particular differentiation among the terms nominated by each method, did not show a clear advantage over any individual term selection method. Although this issue was not completely resolved and requires further investigation, we turned to the more interesting

approach of using iterative refinement of profiles and varying the term selection methods used in the iterations.

Thus, in our TREC-6 experiments we applied the basic profile training procedure (Steps 1–3 in Section 3.1) *twice* over the training data. We expected that the second pass of training would result in higher coverage of terms relevant both to the topic and the training documents and, thus lead to better selection of relevant documents in the target corpus. This was based on the assumption that the first-pass profile provides a better representation of the topic than the original query and that a second pass using the expanded profile will yield even better ranking and selection of training subdocuments in the second phase.

Furthermore, since terms are selected from examples of relevant documents, term specificity will be high, especially because we use phrase-based indexing. However, *controlling* specificity is a critical issue, since the training methods naturally tend to over-generalize based on the set of training examples (see, however, our TREC-6 filtering experiments, Section 4.1.1, in which we made our first attempts at coping with over-fitting effects). In the following section we present the results of our multi-pass training work and describe in detail the official run (CLCOMB) that illustrates this method (see Section 3.4).

Results of our preliminary experiments also indicated that some term-selection methods were more successful for some topics than for others. This observation motivated our effort to identify the best term-selection method for each individual topic, which led to our second official submission, CLMAX (see Section 3.5).

As a general note, our TREC-6 profiles varied in length from run to run. In some experiments we used profiles with fewer than 100 terms, while in others we used profiles with as many as 500 terms. The profiles were generally created from relevant subdocuments found in the top 8,000 subdocuments retrieved from the training data in response to the initial profile or the first-pass profile. For profile training and refinement, we typically used subdocuments containing 12 sentences on average.

3.4 Experiments with Multi-Pass Training – CLCOMB

Comparison of Single-Pass Training with Two-Pass Training

We performed a large number of experiments in an attempt to gain some insight into the complex issue of the interaction between sources of terms and sources of term selection statistics for the two-phase training process. For illustration, we offer here the results of some of our experiments in which we used CLProb in the first pass and one of either CLProb, CLThes, or Rocchio in the second pass. We observe that the CLProb term selection method consistently shows good overall performance in single-pass training and, therefore, represents a good starting point for further profile refinement.

Table 4 provides details of the two-pass training process for selected experiments. Here the specified training database is the source of both terms and term selection statistics. All the experiments use the TREC-5-FBIS database as the source of reference statistics used for relevance matching.

The results of the experiments that use the ExtendedFBIS database in the second pass show no significant contribution of the second-pass training to routing performance on the TREC-6 test data.

Experiment	Term Selection Method	Training DB	Starting Profile	Num of Terms Added
CLProb-Init (one-pass train.)	CLProb	ExtendedFBIS	TREC Topic	300
Ext-2nd CLThes - **CLCOMB**	CLThes	ExtendedFBIS	CLProb-Init	500
Ext-2nd CLProb	CLProb			
Ext-2nd Rocchio	Rocchio			
FBIS-2nd CLThes	CLThes	TREC-5-FBIS	CLProb-Init	200
FBIS-2nd CLProb	CLProb			
FBIS-2nd Rocchio	Rocchio			

Table 4. List of two-pass training experiments with CLProb as the first-pass profile training method.

Furthermore, there is no noticeable difference in effectiveness among different term selection methods in the second phase training (see Table 5).

RUN	Recall	AvgPr	PrAt100docs	Exact Pr
CLProb-Init	5012	0.3005	**0.3374**	**0.3450**
Ext-2nd CLThes - **CLCOMB**	4994	0.2961	0.3315	0.3334
(Rel. improv.)	(-0.4%)	(-1.5%)	(-1.7%)	(-3.3%)
Ext-2nd CLProb	**5074**	**0.3029**	0.3302	0.3389
(Rel. improv.)	(1.2%)	(0.8%)	(-2.1%)	(-1.7%)
Ext-2nd Rocchio	4952	0.3021	0.3255	0.3369
(Rel. improv.)	(-1.2%)	(0.5%)	(-3.5%)	(-2.3%)

Table 5. Training over the ExtendedFBIS data in the second phase. Tested on the TREC-6-FBIS data with reference statistics from TREC-5-FBIS data.

On the other hand, the experiments that use TREC-5-FBIS data (instead of ExtendedFBIS) as a source of profile terminology and term selection statistics show that the second-phase training can result in better overall performance and appreciable improvement in average precision (See Table 6).

RUN	Recall	AvgPr	PrAt100docs	Exact Pr
CLProb-Init	5012	0.3005	0.3374	0.345
FBIS-2nd CLThes	4982	0.3123	0.3406	0.3590
(Rel. improv.)	(-0.6%)	(3.9%)	(0.9%)	(4%)
FBIS-2nd **CLProb**	**5109**	**0.3244**	**0.3472**	**0.3597**
(Rel. improv.)	(1.9%)	(**7.9%**)	(2.9%)	(4.2%)
FBIS-2nd Rocchio	5080	0.3241	0.3445	0.3576
(Rel. improv.)	(1.4%)	(7.9%)	(2.1%)	(3.7%)

Table 6. Training over the TREC-5-FBIS data in the second phase. Tested over the TREC-6-FBIS data with reference statistics from TREC-5-FBIS.

We selected the CLCOMB profiles for our official TREC-6 submission because these profiles performed best over the subset of training data, TREC-5-FBIS, that was expected to be most similar to the TREC-6 test data (see Table 7).

RUN	Recall	AvgPr	PrAt100docs	Exact Pr
CLProb-Init	3895	0.3416	0.2977	0.3805
Ext-2nd **CLThes**	**3995**	**0.4026**	**0.3330**	**0.4205**
(Rel. improv.)	(2.5%)	(17%)	(11.9%)	(10.5%)
Ext-2nd Rocchio	3931	0.3247	0.2981	0.3816
(Rel. improv.)	(0.9%)	(-4.9%)	(0.1%)	(0.3%)
Ext-2nd CLProb	3973	0.3682	0.3951	0.3951
(Rel. improv.)	(2%)	(7.8%)	(32.7%)	(3.8%)

Table 7. Training over ExtendedFBIS in the second phase. Tested over the TREC-5-FBIS data.

Testing over TREC-6-FBIS data, however, indicated that a good fit of CLCOMB profiles to the training data created a bias in profile terminology that made the profiles less effective in general – an over-fitting effect. The best performing run among the experiments described above, FBIS-2nd CLProb, outperforms the CLCOMB profiles significantly, as can be seen in Table 8.

RUN	Recall	AvgPr	PrAt100docs	Exact Pr
Ext-2nd CLThes – **CLCOMB**	4994	0.2961	0.3315	0.3334
FBIS-2nd **CLProb**	**5109**	**0.3244**	**0.3472**	**0.3597**
(Rel. improv.)	(2%)	(9.6%)	(4.7%)	(7.7%)

Table 8. Comparison of CLCOMB and the best results of second phase training over TREC-5-FBIS. Tested over the TREC-5-FBISdata.

Conclusions

Based on the experiments performed to date, we observed no notable improvement of routing performance with combinations of different term selection methods in two-phase profile training. The method that has been generally superior to others, CLProb, is likely to yield the best routing performance when applied twice over the training data.

It is, however, important to note that many parameters are involved in the experiment environment and we have not yet fully understood their subtle interactions. Moreover, the average values of performance measures that we use to characterize the results of the experiments do not allow us to assess the effectiveness of two-phase training for individual topics. In the following section we discuss the possibility of using a two-pass training procedure with various combinations of term selection methods to optimize the routing performance for individual topics.

3.5 Profile Optimization for Individual Topics – CLMAX

Experiment CLMAX illustrates our approach to optimizing routing performance for individual topics by selecting, for each topic, the profile that maximizes routing performance over the training data, ideally data that have not been used for profile generation and refinement.

CLMAX profile sets consist of the profiles from three two-pass training experiments: FBIS-2nd CLThes, FBIS-2nd CLProb, and FBIS-2nd Rocchio (see Table 6 for details). For each topic, we selected the profile that performed best over the TREC-5-FBIS data.

In Table 9 we see that the CLMAX run does not show any improvement over the individual runs. Indeed, had we been able to select optimal profiles for each individual topic against the TREC-6 data, the average precision of the CLMAX run would have been 0.34, thus higher than the average precision of 0.32 achieved by our best performing run, FBIS-2nd CLProb.

RUN	Recall	AvgPr	PrAt100docs	Exact Pr
CLMAX	5041	0.3146	**0.3487**	0.3536
FBIS-2nd CLThes	4982	0.3123	0.3406	**0.3590**
FBIS-2nd CLProb	**5109**	**0.3244**	**0.3472**	**0.3597**
FBIS-2nd Rocchio	5080	0.3241	0.3445	**0.3576**

Table 9. Performance of CLMAX and three individual runs used to select profiles for CLMAX over theTREC-6-FBIS data.

Conclusions

Our CLMAX experiment provides some evidence in support of our original hypothesis that different topics are modeled more effectively by different combinations of term selection methods. The main issue is the design of a method that would enable the system to determine automatically which combination is most appropriate for a given topic. To design such a method the following issues need to be addressed:

(1) the stability across databases of the performance ranking for the set of training methods;
(2) the significance in the difference in performance levels among various methods;
(3) the tendency toward over-fitting to the training data; and
(4) the inherent correlation between the topic and preferred term selection method.

However, we expect to see an improvement in performance from optimization for individual topics. We intend to address such issues in our future research.

3.6 Comparative Performance Analysis of CLARIT TREC-6 Routing

In this section we briefly compare our official submissions, CLCOMB and CLMAX, to those of other participating groups.

Overall, CLMAX performed better than CLCOMB. Compared to other systems, CLMAX achieved an average precision at or above the median for 32 of the 47 routing topics. (See Figures 4 and 5 and Table 10.)

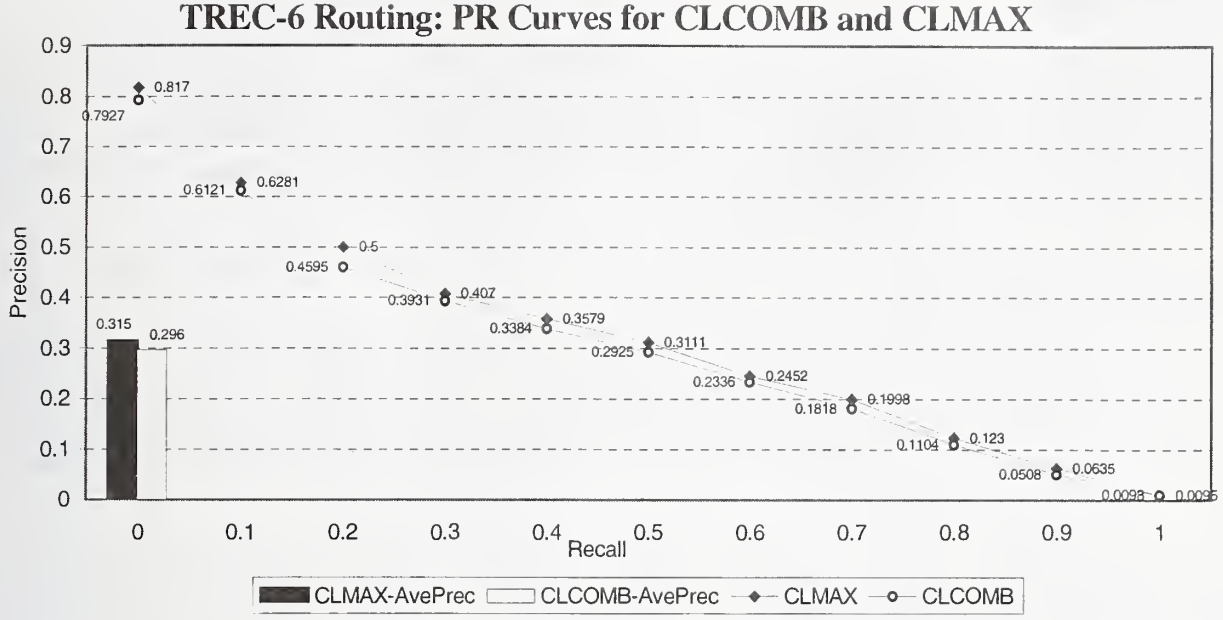


Figure 4. Precision/recall statistics for CLCOMB and CLMAX — CLARIT official routing submissions.

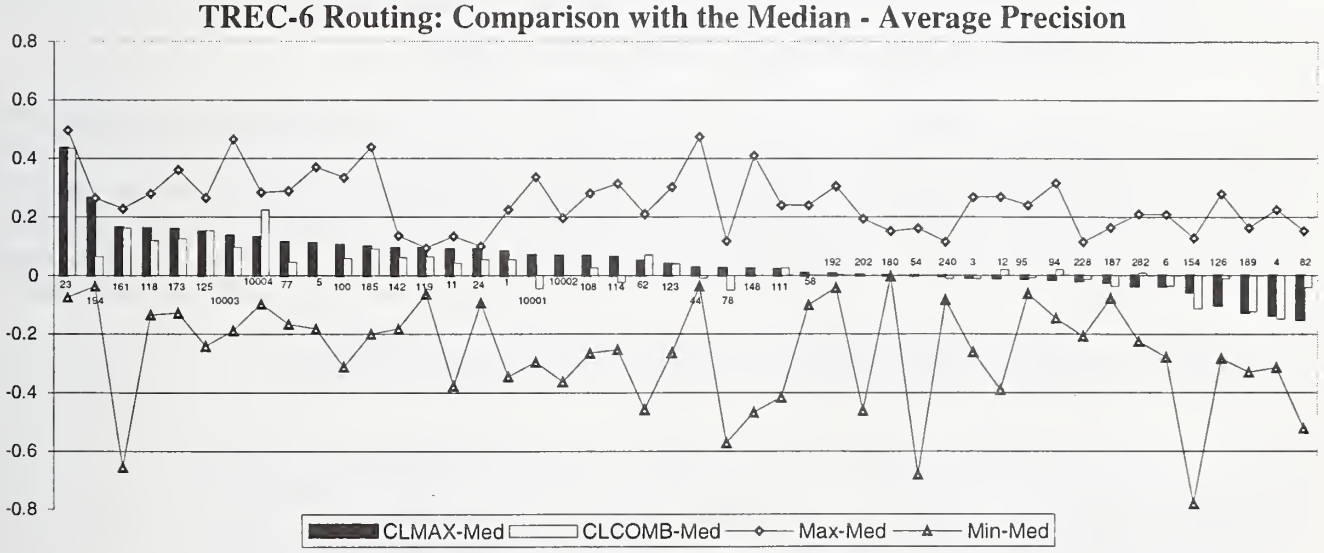


Figure 5. Query-by-query performance of CLCOMB and CLMAX vs. the group median.

RUN	(> Med)	(=Med)	(< Med)	Recall	AvgPr	PrAt100docs	Exact Pr
CLMAX	30	2	15	5041	0.3146	0.3487	0.3536
CLCOMB	25	2	20	4994	0.2961	0.3315	0.3334

Table 10. Comparison of the achieved average precision with the median for individual topics.

4 Filtering

Like most participating groups, we treat the TREC filtering problem as consisting of two separate steps: (1) maximizing the ranking of the routed documents as evaluated by the average precision metric, then (2) finding the utility-specific optimal cut-off on the resulting ranked list. Our official filtering submissions, CLROUTE and CLCOMM, illustrate our efforts to address issues related to both ranking and relevance thresholding.

4.1 Experiment Design

The CLROUTE experiment was focused on optimizing document ranking. It essentially represents an extension of our efforts in multi-pass profile training. It uses the CLProb term selection method in the first pass followed by Rocchio term selection in the second. Following the filtering task guidelines, it uses only FBIS training data as the source of potential profile terms, term selection statistics, and reference statistics.

The method we used to create CLCOMM profiles attempts to address the problem of profile over-fitting to the training data. This phenomenon has a two-fold effect on document filtering: first, it is likely to create a bias in the ranking of test documents and, second, it can result in relevance scores that are significantly different (lower) from those observed in training documents. Bias in document ranking often results in inferior routing performance over new documents, thus directly affecting the "ceiling" on utility values. The difference in relevance score scales, on the other hand, is an obstacle to score-based cut-off thresholds. Predicting optimal threshold values based on the training data becomes highly unreliable under such circumstances.

For thresholding, we compare two different methods. The more straightforward method uses routing over training data to determine the best cut-off point for each individual topic and each utility measure. This cut-off point is then used as a score threshold for the test data ("raw score"). The second method uses logistic regression to model the probability of relevance over the training data as a function of relevance score. For utility measures for which the theoretically optimal cut-off probability can be determined, this cut-off value is then mapped onto a score threshold value.

4.1.1 Document Routing: Profile specificity and score comparability

Typically, profiles tend to contain terms specific to the training corpus. Unless the topic coverage of profile terminology is low because of the nature of the term selection method, the presence of specific terms generally does not affect the system's ability to identify relevant documents (see Section 2 for relevance matching). However, the presence of such terminology makes it difficult to predict the performance of profiles over new documents. Indeed, the degree of overlap between profile and document terminology can differ substantially for training and test data, which in turn can result in significant differences in the ranges of relevance scores for the two data sets. This phenomenon presents an important issue when setting relevance thresholds in filtering.

The CLCOMM experiment represents our attempt to reduce the level of overly specific terminology in topic profiles and thereby achieve better score comparability between training and test data. For that purpose, we divided the FBIS training data into two parts and used each part to generate a profile for each topic (compare with the partitioning results in [Robertson et al. 1997]). We created the final profiles by merging the two initial ones and retaining only common terms. We

hypothesized that such terminology would represent more general, yet highly characteristic features of the topic (see Figure 6).

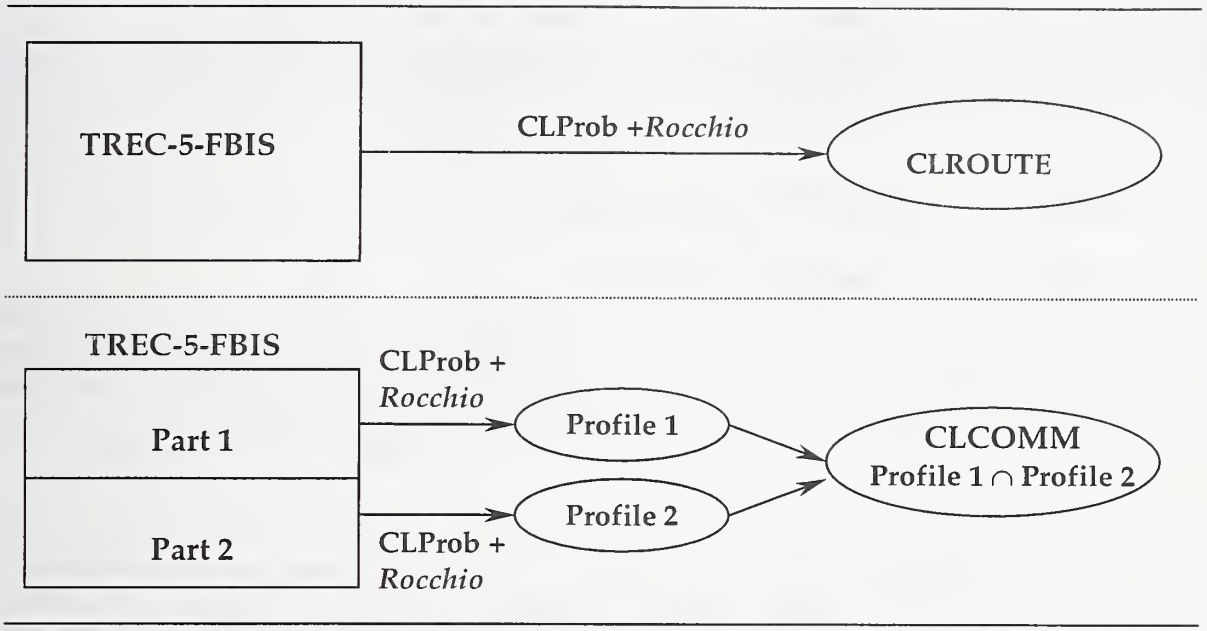


Figure 6. Profile training in the CLARIT filtering experiments: CLROUTE and CLCOMM.

4.1.2 Thresholding Mechanisms

For the threshold setting problem, we explored two methods: (1) thresholds based on raw scores (SC) and (2) thresholds based on estimates of probability of relevance obtained via logistic regression (LR).

For any given utility, the raw-score-based approach simply finds the score cutoff that maximizes that utility over the training set. This same cutoff point is then used as a threshold on the testing documents.

The logistic regression approach, on the other hand, first identifies the relation between scores and the probability of relevance based on the relevance scores of the training documents and their associated relevance judgments (0 or 1) [Hosmer & Lemshow 1989]. Then, whenever the theoretically optimal threshold probability is known and expressible as a constant – as it is for F1 (0.4) and F2 (0.2) – this probability is transformed to a threshold on the raw score by the inverse logistic transformation.

Although probabilistic modeling of document relevance using logistic regression still has the problems of document ranking bias and score comparability across corpora, this approach more successfully addresses the problem of selecting a score threshold for the theoretically optimal utility value. Indeed, utility measures, as functions of document relevance scores, or more precisely, document ranking, are likely to achieve local maxima with slight variations in value over a range of relevance scores. Selecting the score threshold to optimize the utility measure therefore is difficult. Probabilistic modeling of document relevance circumvents this problem. It establishes a smooth mapping between raw scores and the probability of relevance, and thus enables us reliably to select

a score threshold that corresponds to the probability of relevance that optimizes the utility measure, when such probability exists.

Our preliminary experiments indicated that the logistic-regression cutoff performed better than raw-score-based cutoff for utility F1, hence we used logistic regression for all our final F1 submissions. However, for F2 the SC method tended to perform slightly better and we adopted SC for our final F2 submission. For ASP it proved hard to express the cutoff condition in the LR model, hence we used the raw-score approach for all our final ASP submissions.

4.2 Discussion of TREC-6 Filtering Results

4.2.1 Comparison of the Thresholding Methods

The observations we made during our pre-TREC experiments are mostly confirmed by the official results of the CLROUTE experiment on the TREC data: logistic regression performed better than our raw score method for utility F1, while SC resulted in slightly better thresholds for utility F2. For CLCOMM, however, the results were comparable for both thresholding methods[1] (see Table 11).

	Utility	Comparison of Achieved Utility Values for Individual Topics		
		#topics(LR>SC)	#topics(LR=SC)	#topics(LR<SC)
CLROUTE	F1	**24**	13	10
	F2	16	9	**22**
CLCOMM	F1	*18*	12	*17*
	F2	*19*	11	*17*

Table 11. Comparison of the logistic regression cutoff (LR) and raw score cutoff (SC) performance.

4.2.2 Filtering Performance of CLROUTE and CLCOMM

Comparison of performance statistics shows that CLROUTE profiles achieved higher values for recall and average precision than the CLCOMM profiles. In fact, as it turns out, the CLROUTE profiles obtained a better routing performance than our best official routing run, CLMAX, on the TREC routing data (see Table 12).

	Recall	AvgPr	PrAt100	ExactPr
CLROUTE	**5303**	**0.3343**	**0.3600**	**0.3730**
CLCOMM	4837	0.2701	0.3166	0.3087
CLMAX	5041	0.3146	0.3487	0.3536

Table 12. Routing performance statistics for CLROUTE and CLCOMM.

As a direct consequence of the higher overall routing performance of CLROUTE, its "ceilings" (i.e., the maximum achievable values for the F1, F2, and ASP scores, which are a function of document

[1] This is true for a simple count of comparisons of utility values. The *average utility values* on the other hand exhibit a similar trend for CLCOMM as for CLROUTE, namely that LR is better for utility F1, and SC is slightly better for utility F2.

ranking) are higher than the corresponding ceilings for CLCOMM. This is amply confirmed in Table 13, which shows the number of topics for which the CLROUTE utility ceiling is greater, equal, or lower than its counterpart for CLCOMM.

Measure	Cutoff	#(CLRoute>CLComm)	#(CLRoute=CLComm)	#(CLRoute<CLComm)
F1	Optimal	33	7	7
F2	Optimal	39	5	3
ASP	Optimal	42	1	4

Table 13. Per topic comparison of the utility value "ceilings" for CLROUTE and CLCOMM.

Furthermore, comparison of utility values achieved in the two experiments (see Table 14), shows that CLROUTE clearly outperformed CLCOMM for the utility measures F1 and F2. For ASP, on the other hand, CLCOMM achieved similar performance.[2]

In fact, a detailed analysis of the two runs leads to the following observations.

First, the performance statistics in Table 14 reveal no significant difference in the relative performance of the two runs when either cutoff method is applied for a given utility measure. Thus, our selection of cutoff technique, logistic regression for F1 and raw-score cutoff for F2, is not a factor in the lower performance of the CLCOMM run.

Measure	Cutoff	#(CLRoute>CLComm)	#(CLRoute=CLComm)	#(CLRoute<CLComm)
F1	**LogReg**	**23**	**12**	**12**
	Score	21	10	16
F2	LogReg	29	5	13
	Score	**31**	**5**	**11**
ASP	Score	20	7	20

Table 14. Per topic comparison of the achieved utility values for CLROUTE and CLCOMM.

Second, a correlation analysis confirms that the improvement of CLROUTE over CLCOMM is largely due to the increase of the ceiling for utilities F1 and F2, whereas for ASP there is no correlation at all between ceilings and actual values.

Finally, even though CLCOMM appears to move closer towards score comparability between training and testing sets than CLROUTE – and also more often actually hits the optimal threshold – a comparison between actual threshold value and optimal threshold value shows that the threshold is still generally set too high (see Tables 15 and 16).

[2] This is again based on a simple count. In fact, CLCOMM performs better than CLROUTE when we compare their average ASP values.

	Utility	Cut-off Method	Too High	Optimal	Too Low
CLROUTE	F1	LogReg	30	3	14
		Score	30	2	15
	F2	LogReg	37	1	9
		Score	35	1	11
	ASP	Score	40	0	7

Table 15. Per topic comparison of the achieved and the optimal threshold value for CLROUTE.

	Utility	Cut-off Method	Too High	Optimal	Too Low
CLCOMM	F1	LogReg	29	5	13
		Score	25	5	17
	F2	LogReg	31	3	13
		Score	28	2	17
	ASP	Score	35	0	12

Table 16. Per topic comparison of the achieved and the optimal threshold value for CLCOMM.

4.3 Comparison with Other Groups

Table 17 compares our official runs with the medians of all the groups and shows our results to be above or equal to the median for most topics for each run.

ProfileVector	Measure	ThreshMethod	(>med)	(=med)	(<med)	(>=med)
CLROUTE	ASP	Score	16	13	18	**29**
CLCOMM			23	6	18	**29**
CLROUTE	F1	LogReg	31	8	8	**39**
CLCOMM			24	9	14	**33**
CLROUTE	F2	Score	27	9	11	**36**
CLCOMM			22	7	18	**29**

Table 17. Comparison of the achieved utility values with the group median for individual topics.

When considering individual runs, CLROUTE and CLCOMM were ranked fifth and seventh, respectively, (out of 17 submissions) for F1 and F2, and ninth and eighth, respectively, for ASP (out of 15).

In comparison with other TREC-6 Filtering systems, the CLARIT system was ranked third for F1 and F2, behind the AT&T [Singhal 1998] and City University [Walker et al. 1998] filtering systems in both cases, and fifth for ASP. A detailed analysis of the filtering tasks and results, including a discussion of the significance of system rankings and the statistical differences in performance of submitted runs and participating groups, is provided by David Hull elsewhere among the collection of TREC-6 reports.

5 CLARIT Ad-Hoc Retrieval

5.1 Experiment Design

The CLARIT TREC-6 Ad-Hoc Retrieval experiments follow essentially the design of the TREC-6 High Precision Task. In particular, the CLARIT Interactive System was used for interactive search over the target corpus to collect, for each topic, samples of documents that are relevant and non-relevant to the topic, as judged by the CLARIT user. The relevance judgements were subsequently used in the batch retrieval process to enhance the query via CLARIT automatic query expansion techniques.

The differences in design between the High Precision Track and the CLARIT Ad-Hoc experiments are in the requirements on the user's interaction with the system. While the High Precision Track guidelines impose a time limit (5 minutes per topic) on the user's interaction with the system, no such restriction was imposed in the CLARIT Ad-Hoc experiments. Instead, the CLARIT users were instructed to search for 10 to 20 relevant documents per topic and, along the way, mark non-relevant documents that could be useful for negative feedback. On the other hand, in the High Precision Track no restrictions were imposed on the type of interaction between the user and the system. In the CLARIT Ad-Hoc experiments, interaction was restricted to manual query construction and modification and to the review of retrieved documents. More precisely, the CLARIT users were asked to formulate initial queries from original topic descriptions in the form of a CLARIT compound query, i.e., a natural language query optionally supplemented with Boolean type constraints when appropriate (see Figure 7). They could read the titles of retrieved documents or view document text to assess document relevance. Based on inspected documents, the users could manually modify the query: reweight query terms, add or delete query terms, or modify query constraints. However, no system-assisted query modification was permitted during the interactive search.

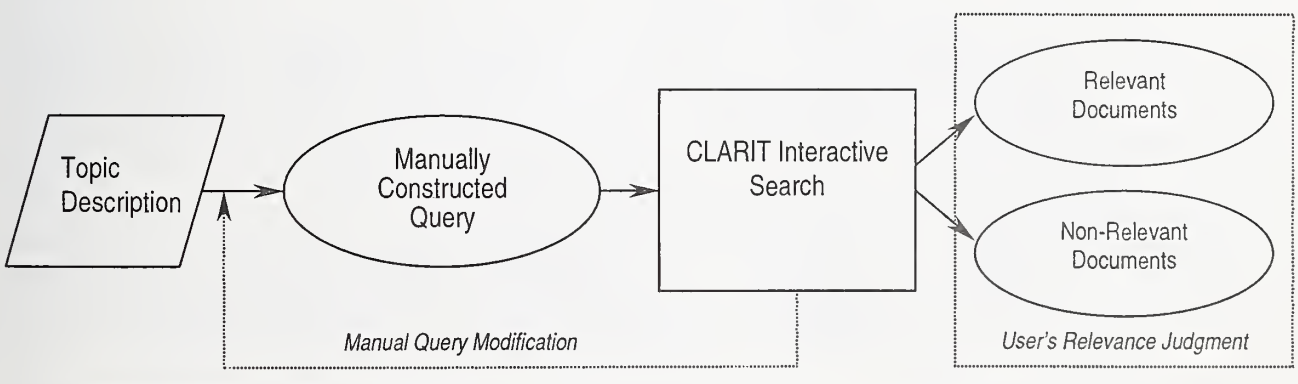


Figure 7. Interactive query formulation and relevance assessment.

The CLARIT users involved in the interactive search included four of the authors of this report, one CLARITECH linguist, and two non-technical volunteers. The 50 ad-hoc topics were divided up approximately equally among the seven users. Most users created the initial query by taking the raw text of the TREC topic as the natural-language statement and specifying at most one or two constraints, if any. Review of retrieved documents was focused on the top-most ranked documents; users scanned documents quickly until they felt able to render a relevance judgment. Users spent about twenty

minutes per topic, on average. The queries and the users' relevance judgments were recorded and then passed on to an automatic batch-mode retrieval process. For the 50 topics, 641 positive judgements and 1,352 negative judgments were recorded. Table 18 gives statistics on the users' judgments.

	Positive	Negative
Total	641	1,342
Average	12.82	26.84
SDV	9.85	37.81
Maximum Num for one topic	46	205
Minimum Num for one topic	0	0
Number of topics w/o judgments	1	9

Table 18. Statistics about CLARIT users relevance judgments.

We submitted two TREC-6 manual ad-hoc runs: CLREL and CLAUG. The purpose of these experiments was (1) to test the effect of user feedback on ad-hoc retrieval, and (2) to explore the effectiveness of the two-pass term-selection technique for automatic query expansion, combining user relevance feedback and system provided (pseudo-relevance) feedback. The CLREL run used relevance judgments from CLARIT users to augment the initial queries automatically (see Figure 8).

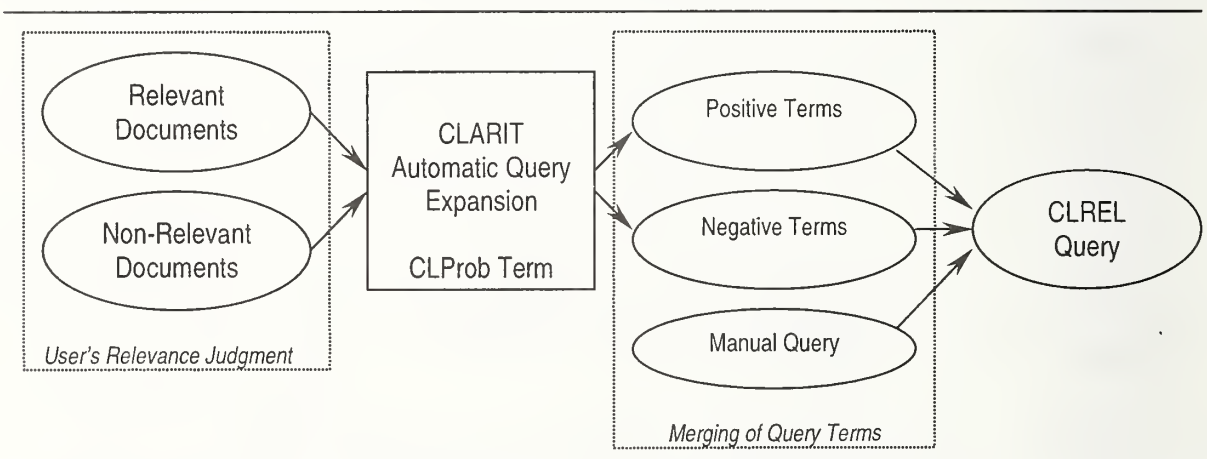


Figure 8. Construction of CLREL queries.

The CLAUG run involved another pass of query refinement using pseudo-feedback, i.e., feedback using the top *N* ranked subdocuments, as determined by the CLREL results (see Figure 9).

In CLREL, we used the CLProb term selection method to identify terms for query expansion. The positive examples (documents judged relevant) were used to extract positive terms; negative examples (documents judged non-relevant) were used to extract negative terms. Fifty positive and thirty negative terms for each topic were merged with the manual queries to create revised query vectors.

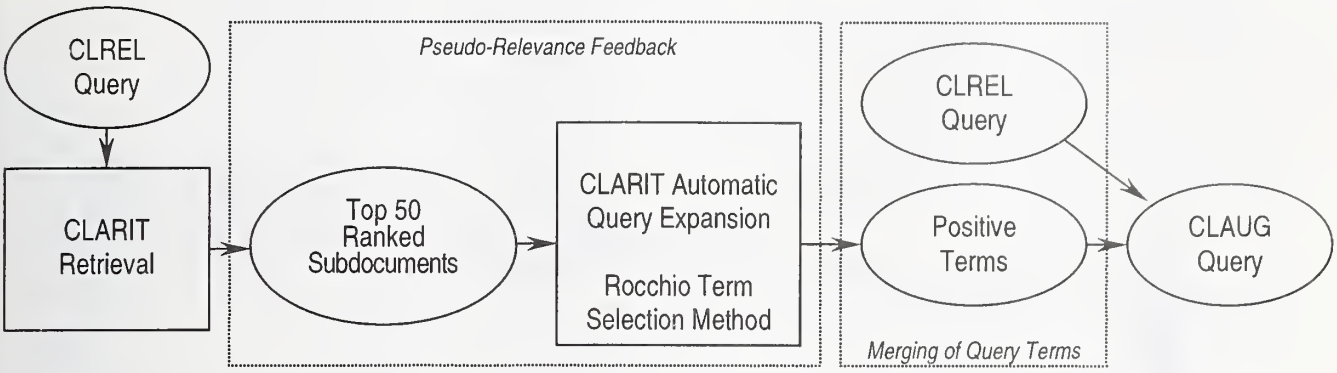


Figure 9. Construction of CLAUG queries.

In the CLAUG experiment, we used a second feedback loop to enhance the query vector formed in the CLREL process. We used Rocchio scoring over the top 50 subdocuments retrieved by the CLREL final vector to identify supplemental query terminology. The top 50 subdocuments were selected fully automatically, excluding those subdocuments that came from documents previously marked as non-relevant by the users. Rocchio scoring was here used only for term selection, not for query term weighting or re-weighting. User specified constraints, which are a part of the manually constructed queries, were used in the initial retrieval phase to facilitate the selection of subdocuments for automatic feedback. However, they were not used in the final CLAUG retrieval.

5.2 Performance Analysis

Figure 10, Table 19, and Table 20 give the results of the CLREL and CLAUG runs in terms of absolute performance. Figure 11 and Table 21 give CLARIT results in comparison to group aggregate performance.

As we noted earlier, the CLREL and CLAUG experiments are very close in design to the High Precision Track experiments (see Section 5.1). Although they are not fully comparable with the TREC-6 High Precision runs, it is interesting to note that the techniques used in these experiments lead to very good performance on the measures adopted by that track. For example, considering the "Precision at 10 Docs", which is one of the main evaluation measures in the High Precision Track, we note that the CLAUG run outperforms all other systems by achieving a precision of 0.612 (see Table 21). Similarly, the CLREL run with a precision of 0.596 comes close to the top performing run by the Cornell/SabIR Research (SMART) system, which achieved a precision of 0.602. (See the reports for the High Precision Track for TREC 6.)

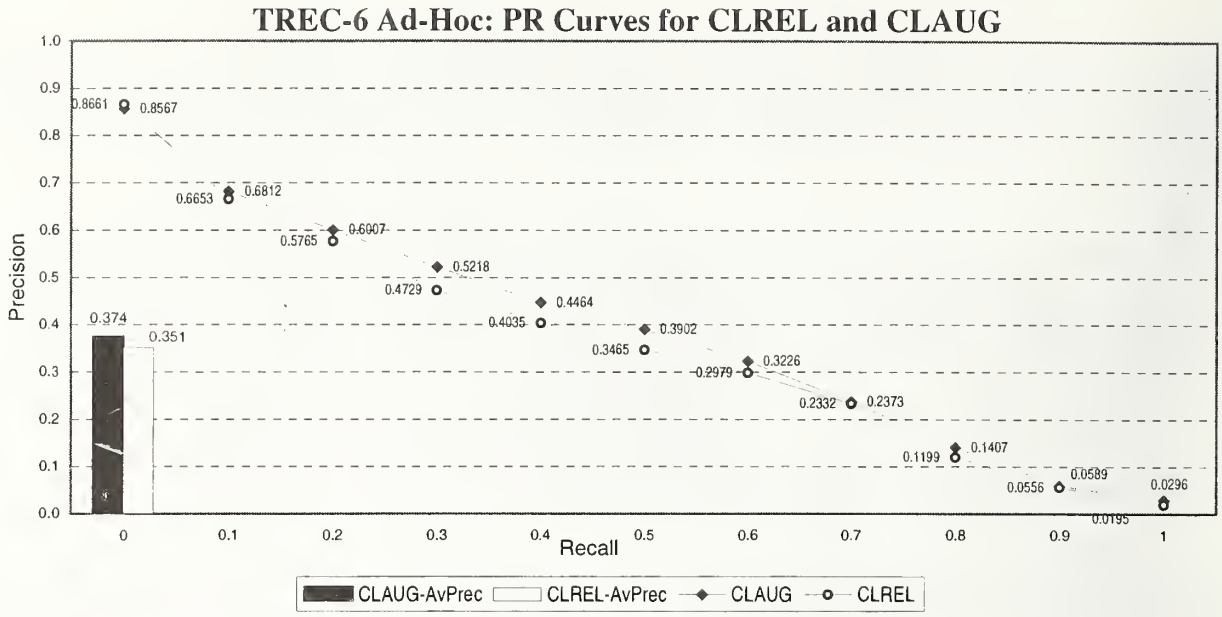


Figure 10. Precision/recall curves for CLREL and CLAUG — CLARIT official ad-hoc submissions.

Run	Recall	Avg.Precision	Initial Precision	Exact Precision	Pre. 100 docs
CLREL	2998	0.3514	0.8711	0.3639	0.2712
CLAUG (over above)	3095 (+3.24%)	0.3742 (+6.52%)	0.8567 (-1.09%)	0.3914 (+7.56%)	0.2822 (+4.05)

Table 19. Performance statistics for CLREL and CLAUG — CLARIT official ad-hoc submissions.

Document Level Averages

CLREL		CLAUG	
At 5 docs	0.7000	At 5 docs	0.7120
At 10 docs	0.5960	At 10 docs	0.6120
At 15 docs	0.5280	At 15 docs	0.5493
At 20 docs	0.4910	At 20 docs	0.5080
At 30 docs	0.4340	At 30 docs	0.4620
At 100 docs	0.2712	At 100 docs	0.2822
At 200 docs	0.1834	At 200 docs	0.1898
At 500 docs	0.1007	At 500 docs	0.1037
At 1000 docs	0.0600	At 1000 docs	0.0619
Exact Precision:	0.3639	Exact Precision:	0.3914

Table 20. Precision at *N* retrieved documents for CLREL and CLAUG.

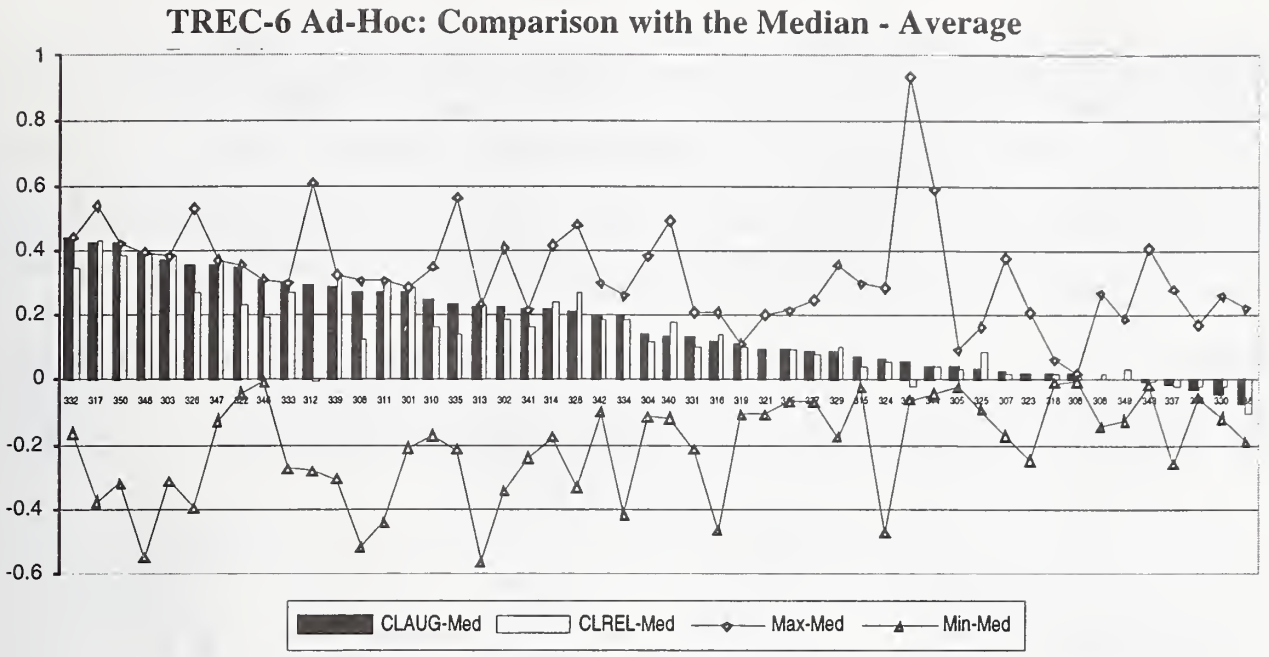


Figure 11. Query-by-query performance of CLREL and CLAUG vs. the group median.

Run	Average Precision			
	>=median	<median	=best	=worst
CLREL	43	7	7	0
CLAUG	45	5	7	0

Table 21. CLARIT ad-hoc results compared to TREC group performance.

5.3 Effects of Document Relevance Judgments

In our post-TREC experiments we compared the NIST and CLARIT users' relevance judgments and evaluated the relative impact of judgment differences on retrieval performance.

Table 22 summarizes the differences between NIST and CLARIT relevance judgments for the 1,592 documents that were judged by both the NIST judges and CLARIT users for the 50 topics.

		CLARIT		
		Yes	No	*Total*
NIST	Yes	**445**	121	566
	No	170	**856**	1026
	Total	615	977	1592

Table 22. Comparison of the CLARIT users' and NIST's relevance judgments.

The agreement between the two judgments is calculated as:

$$\text{Agreement} = \frac{Number\_with\_same\_judgment}{Total\_judged\_documents}.$$

In our case, this equals (445+856)/(445+856+121+170) or 0.8172. We note that this level of agreement is comparable to the levels reported by NIST in studies of inter-rater reliability among TREC judges.

We conducted two sets of experiments to test the effect of the difference in relevance judgments on the retrieval performance. We compared the official CLARIT Ad-Hoc runs with the results of experiments that use two different document relevance assessments: first, the "corrected" relevance judgments of the CLARIT users and, second, the relevance judgments of the NIST judges.

Experiments NISTREL and NISTAUG use the "corrected" relevance judgments of the CLARIT users: CLARIT users' relevance judgements were revised to reflect the relevance judgments of the NIST judges. In all other respects, NISTREL and NISTAUG are identical to the CLREL and CLAUG runs. Comparison of retrieval performance is given in Table 23.

Run	Recall	Average Precision	Initial Precision	Exact Precision	Precision at 100 docs
CLREL	2998	0.3513	0.8661	0.3639	0.2712
NISTREL (over above)	3029 (+1.03%)	0.4243 (+20.8%)	0.9801 (+13.2%)	0.4252 (+16.8%)	0.2748 (+1.33)
CLAUG	3095	0.3742	0.8567	0.3914	0.2822
NISTAUG (over above)	3128 (1.07%)	0.4484 (+19.8%)	0.9653 (+12.7%)	0.4505 (+15.1%)	0.2914 (+3.3%)

Table 23. Effects of user feedback based on CLARIT users' judgments and "corrected" relevance judgments.

The second set of experiments compares the effectiveness of the relevance feedback based on complete NIST relevance judgments and the original CLARIT users' relevance judgments, respectively. Initial users' queries were enhanced with terms extracted from the top *N* ranked subdocuments that originate from the judged documents. Table 24 shows the results of ad-hoc experiments that use the top 50 and top 100 ranked subdocuments in the feedback loop; Figures 12 and 13 amplify the details.

Run	Recall	Average Precision	Initial Precision	Exact Precision	Precision at 100 docs
CLARIT-50	2861	0.3376	0.8543	0.3585	0.2518
NIST-50 (over CL-50)	2940 (+2.76%)	0.4170 (+23.5%)	0.9801 (+14.7%)	0.4144 (+15.6%)	0.2656 (+5.48)
CLARIT-100	2966	0.3389	0.8521	0.3592	0.2574
NIST-100 (over CL-100)	3072 (+3.57%)	0.4629 (+36.6%)	0.9789 (+14.9%)	0.4587 (+27.7%)	0.2894 (+12.4%)

Table 24. Effects of the user feedback based on the CLARIT users' and NIST's relevance judgments – judgments limited to the top *N* retrieved documents.

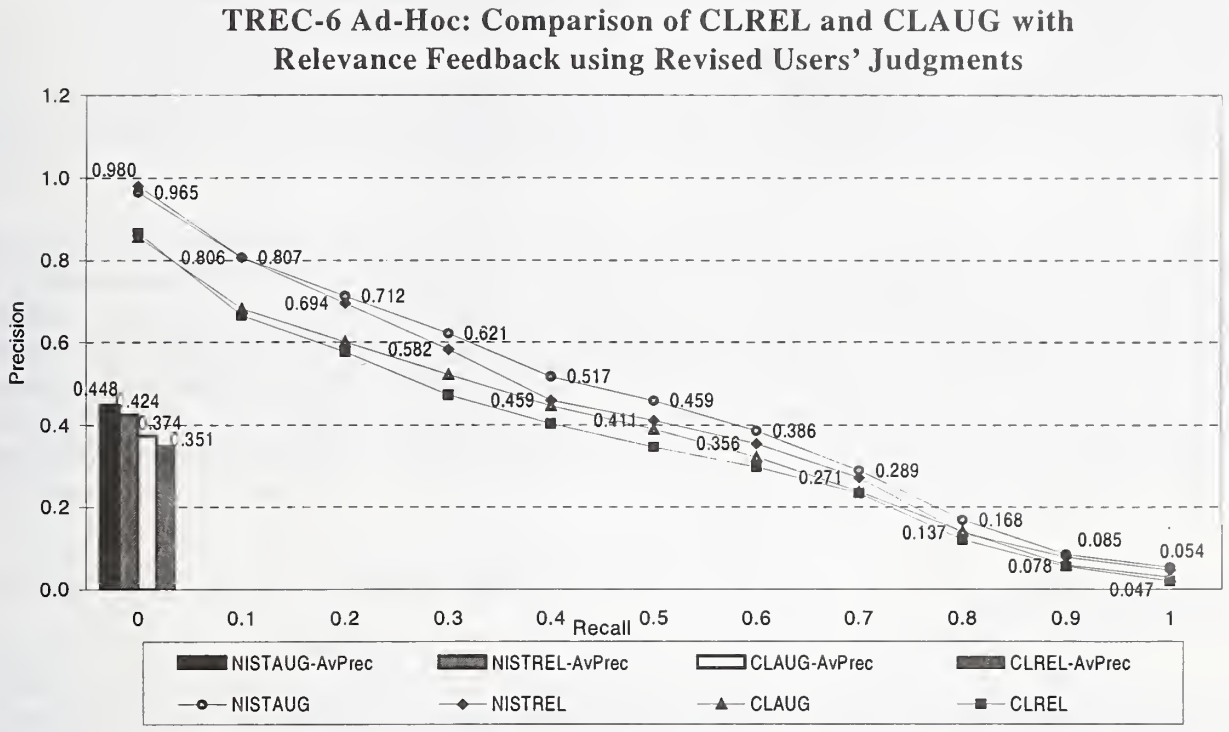


Figure 12. Effects of relevance feedback based on revised CLARIT users' judgments.

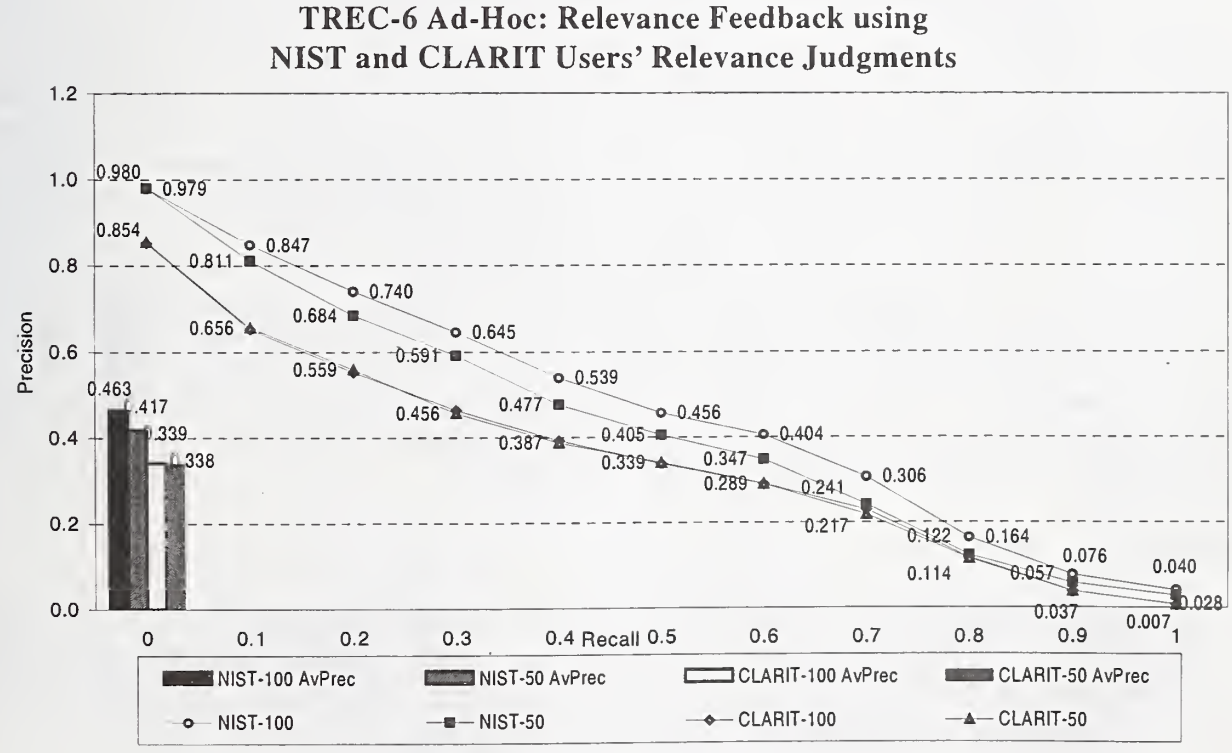


Figure 13. Effects of relevance feedback based on NIST's relevance judgments.

In general, we can see that "correct" relevance judgments have a dramatic impact on the performance of the system. Of course, from the point of view of the CLARIT users, all the documents they marked as relevant were "correct". Thus, depending on one's point of view, this evaluation can be regarded as giving a practical upper limit on the performance of the system (the results with NIST judgments substituted selectively for CLARIT user judgments) or a measure of the distortion introduced by conflicting user judgments (approximately 20% in average precision – see Table 23), which cannot be avoided in actual retrieval applications.

It is interesting to note that, with a relatively small number of "correct" judgments per query (cf. NIST-50 and NIST-100), the CLARIT system attained a level of performance equivalent to that of the system with the highest ranking performance in TREC 6, viz., the University of Waterloo system. The University of Waterloo [Cormack et al. 1998] approach depends on exhaustive interactive searching, requiring many days of effort. In contrast, the CLARIT system results show that our approach is equally effective (in the hands of a "true" judge) when the user renders only a limited number of judgments, as is the case in realistic time-limited interactive search. We regard our results as a strong validation of our approach.

6 Chinese Track

The main objective of our TREC-6 Chinese experiments was to establish a reasonable baseline for future work in which we intend to explore more elaborate indexing and query processing methods. In particular, we hope to revisit the issue of effectiveness of automatic pseudo-feedback for Chinese texts.

Our baseline Chinese retrieval system used overlapping character bigrams as the basis for indexing documents and parsing queries. We thus avoided the need for specialized linguistic resources such as Chinese lexicons and grammars. Although we had already applied this approach in some of our TREC-5 Chinese experiments (see [Tong et al. 1997a]) we wanted to assess the method within the newly developed CLARIT evaluation environment. Furthermore, since we could not devote sufficient time in TREC-5 experiments to the design of an effective feedback procedure for Chinese, we decided to explore this issue in more detail in our TREC-6 work. In particular, in our TREC-5 experiments we used the CLARIT Thesaurus Discovery technique to identify terminology for automatic query expansion. We knew at that time that Thesaurus Discovery would be unreliable because some of the linguistic features we exploit in that technique – requiring identification of phrases and their constituents, which in turn depends on morphological and syntactic analysis – were not available to us in our processing of Chinese text. To address this problem, in our TREC-6 experiments we applied and evaluated several newly developed term-selection methods that are less dependent on linguistic analysis and yield retrieval performance improvements similar to what have observed over English texts with Thesaurus Discovery.

6.1 Experiment Design

For the official submission we selected three runs: two with fully automatic query processing, CLARITcAS and CLARITcAL, and one using manually constructed queries, CLARITcM. The CLARITcAS run used automatic query construction with short descriptions of the TREC topics; CLARITcAL used long descriptions.

The manual queries differed from the automatic ones in that (1) the automatically parsed long queries were edited (minimally) and (2) Boolean type constraints were added. The user concentrated on adjusting the weights on bigrams (terms) in the parsed query vector. Very few additional bigrams were added or removed from the original query vector. Manual query preparation involved no interaction with the target corpus and took a total of approximately three hours for the set of 26 Chinese topics. All query formulation was performed by a single native Chinese speaker.

In all three experiments we used automatic pseudo-feedback to enhance the initial query vectors. In the experiments with automatically generated query vectors, CLARITcAS and CLARITcAL, we applied the CLProb term-selection method to the 50 top-ranked subdocuments from the initial search and added up to 100 bigrams to the query. Similarly, we used the top 75 subdocuments in the experiment with manual queries and added automatically up to 150 bigrams to the original query. The selection of subdocuments for pseudo-feedback in experiments with manual queries was facilitated by the user specified constraints. The constraints were not used to obtain the final set of retrieved documents.

6.2 Performance Analysis – Chinese Retrieval

Figure 14 and Table 25 show the results of our TREC-6 official runs in the Chinese track.

We observe that the manual queries performed slightly better than automatic queries based on long descriptions of the topics. As expected, the performance of short queries is inferior to the other two runs.

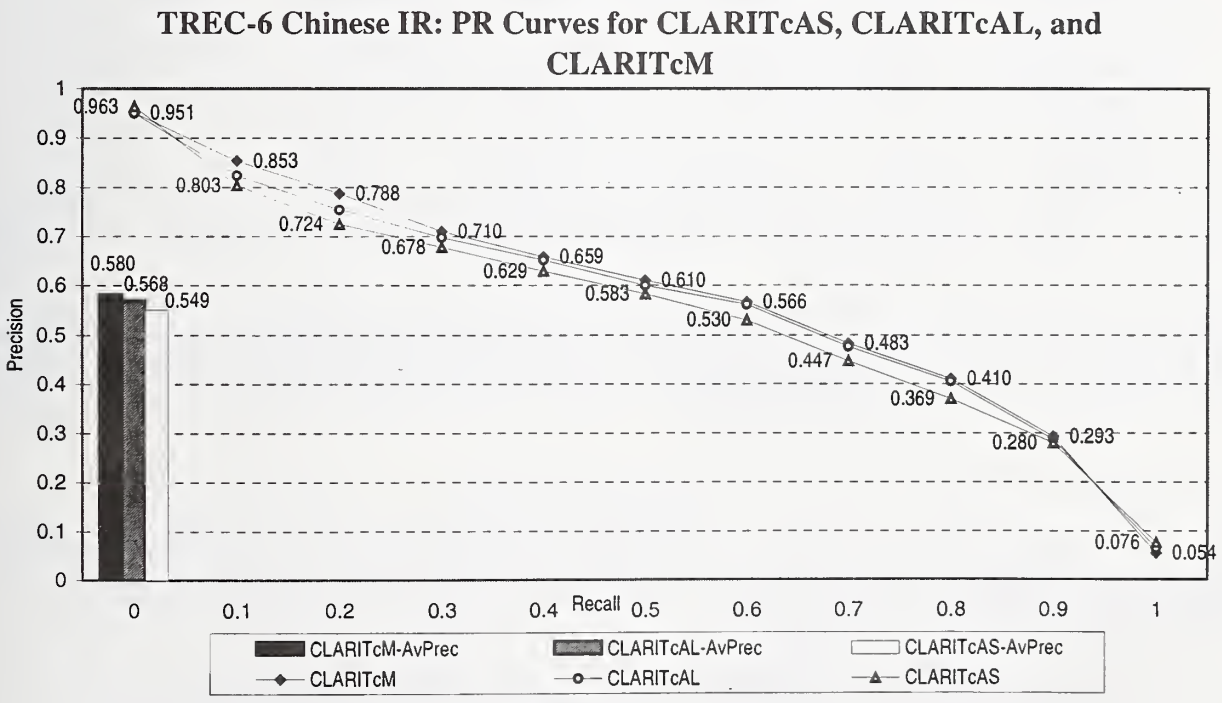


Figure 14. Precision/recall curves for CLARITcAS, CLARITcAL, and CLARITcM — CLARIT official Chinese track submissions.

Run	Recall (total: 2958)	Average Precision	Initial Precision	Exact Precision	Precision at 100 docs
CLARITcAS	2719	0.5495	0.9634	0.5357	0.4938
CLARITcAL (over above)	2746 (+0.99%)	0.5683 (+3.42%)	0.9507 (-1.32%)	0.5464 (+2.00%)	0.5115 (+3.58%)
CLARITcM (over above)	2774 (+1.02%)	0.5797 (+2.01%)	0.9512 (+0.05%)	0.5475 (+0.20%)	0.5242 (+2.48%)

Table 25. Performance statistics for the CLARIT Chinese track experiments.

Comparison of these results with a baseline of simple retrieval without automatic expansion of queries shows consistent improvement on all evaluation measures due to pseudo-feedback. In particular, the average precision shows relative improvement of more than 10% for all three experiments that used pseudo-feedback to enhance the original query vectors (see Table 26).

Run	Recall	Average Precision	Initial Precision	Exact Precision	Precision at 100 docs
Short w/o Feedback	2587	0.4837	0.9515	0.4807	0.4500
CLARITcAS (over above)	**2719** (+5.10)	**0.5495** (+13.60%)	**0.9634** (+1.25)	**0.5357** (+11.44%)	**0.4938** (+9.73%)
Long w/o Feedback	2634	0.5111	0.9304	0.5076	0.475
CLARITcAL (over above)	**2746** (+4.25)	**0.5683** (+11.19%)	**0.9507** (+2.18%)	**0.5464** (7.64%)	**0.5115** (+7.68)
Manual w/o Feedback	2641	0.5209	0.9347	0.5095	0.4727
CLARITcM (over above)	**2774** (+5.04)	**0.5797** (+11.29%)	**0.9512** (+1.77%)	**0.5475** (+7.46%)	**0.5242** (+10.89%)

Table 26. Effects of CLARIT automatic feedback in the Chinese text processing.

Generally, the results of our simple approach to indexing and query processing using overlapping bigrams are very encouraging. We expect to achieve further improvements in retrieval performance by applying more sophisticated, language-specific text-analysis techniques.

6.3 Comparison with Other Participating Systems – Chinese Retrieval

Figure 15 and Table 27 show the performance of the CLARIT Chinese runs in comparison with the median performance for individual topics as calculated for the set of all submitted runs in the TREC-6 Chinese track.

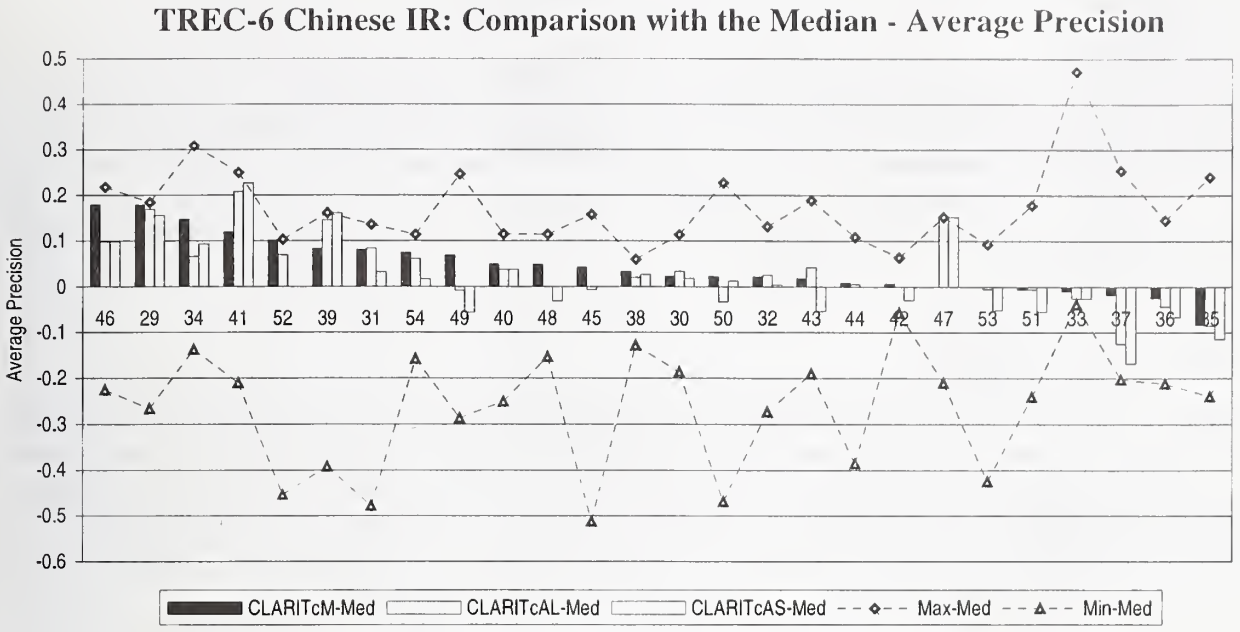


Figure 15. Query-by-query performance vs. the group median.

Run	Average Precision			
	>= median	<median	=best	=worst
CLARITcAS	16	10	2	0
CLARITcAL	17	9	0	0
CLARITcM	21	5	0	0

Table 27. Comparison of the achieved average precision with the median for individual topics.

In particular, we note the excellent relative performance of the CLARIT experiment with manually constructed queries, CLARITcM: the average precision is above the median for 21 out of 26 topics. Since the initial manual queries in CLARITcM are highly similar to the automatically constructed initial queries in CLARITcAL, this analysis underscores the beneficial effects on some of the topics of constraint facilitated pseudo-feedback with larger numbers of terms (up to 150 bigrams per query). This is consistent with the results of our TREC-5 Chinese and Ad-Hoc experiments that apply similar approaches to automatic query expansion [Tong et al. 1997a].

7 Spoken Data Retrieval (SDR) Track

The SDR-track task involved "known item search" – the retrieval of a single known-relevant document for each of 49 topics from a corpus of speech transcripts. The performance measures used in this track are therefore different from the ones used in other TREC tracks. Instead they are based on the *rank* of the retrieved known item, viz., the *average rank* and the *average inverse rank* over the given topics.

The CLARIT SDR track experiments were performed using the CLARIT Retrieval engine over speech transcripts (the control or baseline transcripts) provided to us by the Linguistic Data Consortium (LDC) and speech transcripts generated by the Informedia group at Carnegie Mellon University, Pittsburgh, PA (see [Siegler et al. 1998]). Our experiments were intended to evaluate (a) the robustness of the straightforward CLARIT retrieval over the speech data and (b) the suitability of a query expansion technique that we developed to enhance search over "corrupted" data. We took a similar query expansion approach in the TREC-5 Confusion Track (see [Tong & Evans 1996], [Tong et al. 1997b]).

7.1 Experiment design

In accordance with the SDR track guidelines, we ran retrieval experiments over three sets of data: (1) the reference database with corrected speech transcripts, (2) the baseline speech transcripts that were provided to all participating groups by LDC, and (3) the set of speech transcripts generated by the speech recognition system from the Informedia group. For each set of speech transcripts we performed two experiments, straightforward CLARIT retrieval and CLARIT retrieval with expanded queries (see Table 28).

Run	Database	Query
CLARITR1	Reference	No Expansion
CLARITB1	Baseline	No Expansion
CLARITB2	Baseline	Expansion
CLARITS1	CMU Speech Data	No Expansion
CLARITS2	CMU speech Data	Expansion

Table 28. Description of the CLARIT official TREC-6 SDR runs.

In the experiments with no query expansion, queries were parsed automatically into single words and phrases using CLARIT NLP techniques. In the experiments with query expansion, we expanded the original query vectors with new query terms taken from the target corpus. These new terms were generated by ranking terms in the target corpus according to their similarity to each individual original query term. Term similarity was calculated based on a character substitution matrix learned from the training data. The selected term variants were assigned the same IDF score as the original query term.

7.2 Performance analysis

The retrieval performance of our SDR experiments is summarized in Table 29 and Figure 16. We note a good performance over the reference data with an average rank of 6.67 and average reciprocal rank of 0.81. The performance over speech transcripts from the Informedia group is better than the performance over baseline transcripts with the average rank better by 26.8%, and the average reciprocal rank by 12.8%, for the experiments without query expansion.

The trend is similar in the experiments with query expansion. However, query expansion generally reduces the effectiveness of the retrieval for both the baseline speech transcripts and the transcripts from the Informedia group. This effect is not surprising since the technique relies on statistical analyses of the target corpus or a training corpus and is sensitive to the size of such a corpus

		Experiments without Query Expansion		Experiments with Query Expansion	
Run	CLARITR1	CLARITB1	CLARITS1	CLARITB2	CLARITS2
Ave Rank	6.67	24.67	18.06	29.16	19.9
Ave Rec. Rank	0.8094	0.6453	0.7277	0.6218	0.7245
Known items found at rank:					
<= 5	44	35	40	36	40
<= 10	44	38	42	37	41
<= 20	46	41	45	41	44
<= 100	48	47	47	45	47
Not found	0	0	0	0	0

Table 29. Performance of the CLARIT official TREC-6 SDR runs.

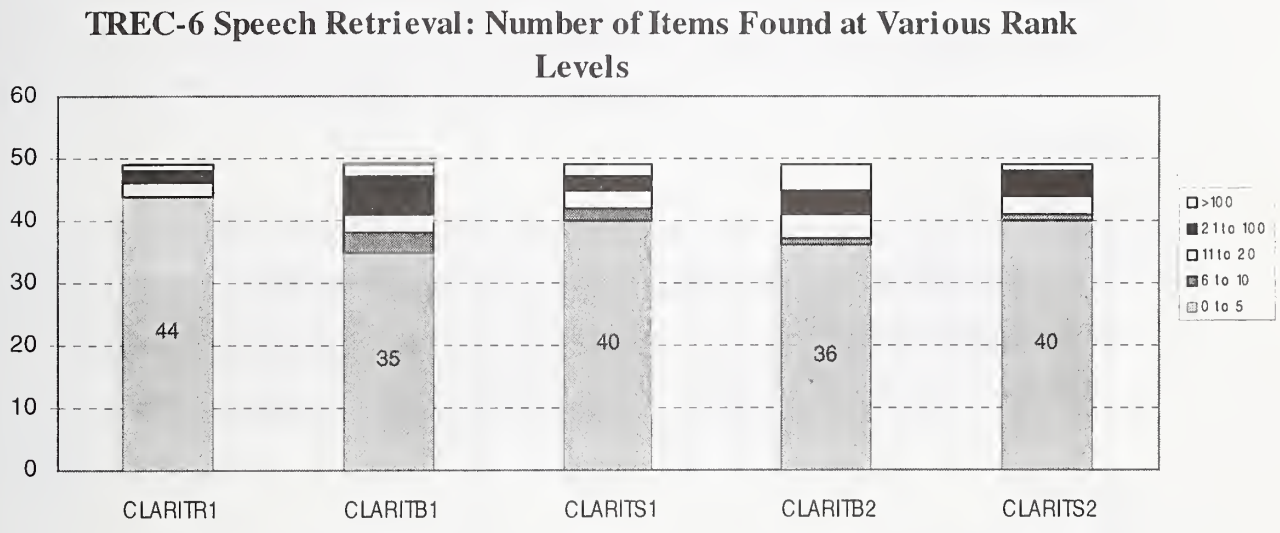


Figure 16. Ranks of the known-items found in the CLARIT official SDR runs.

Compared with other participating systems, the CLARIT System achieved median or better than median rank for a large number of topics, performing best for more than 60% of all topics in experiments without query expansion and more than 55% of all the topics in the experiments with query expansion (see Table 30).

	Rank			
Run	>= median	<median	=best	=worst
CLARITR1	41	8	37	4
CLARITB1	41	8	30	4
CLARITS1	43	6	33	0
CLARITB2	39	10	27	5
CLARITS2	42	7	33	0

Table 30. Per topic comparison of the retrieval rank with the median rank for the individual query.

From Tables 29 and 31 and Figures 17 and 18, which give CLARIT and group aggregate performance results, we see that the CLARIT system performed consistently above or close to the median for the reference transcripts and the Informedia speech transcripts. The performance was slightly worse for the baseline speech transcripts.

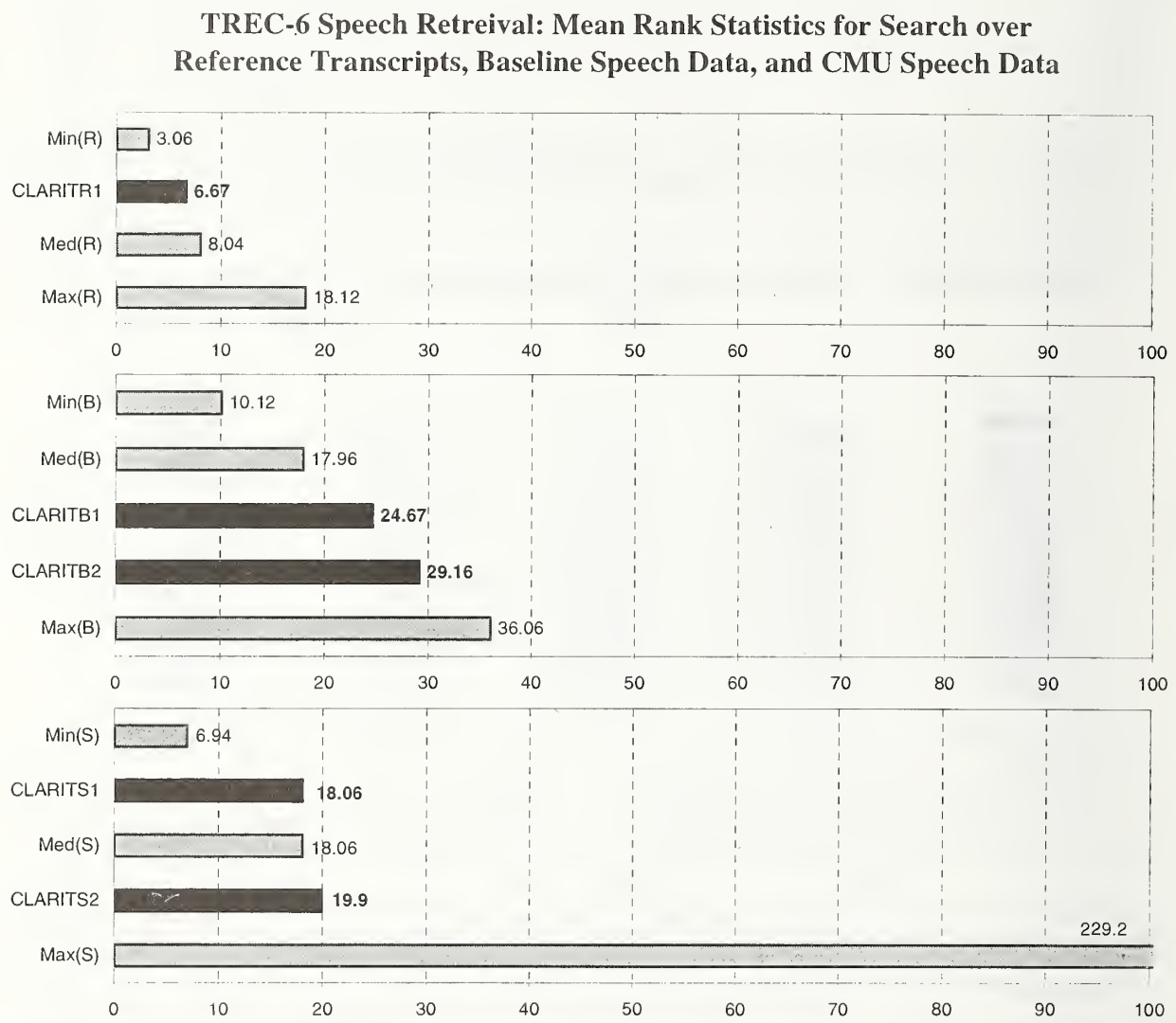


Figure 17. Comparison of the CLARIT average rank with the group median.

	Reference Transcript			Baseline Recognizer			Own Recognizer		
	Min	Med	Max	Min	Med	Max	Min	Med	Max
Ave Rank	3.06	8.04	18.12	10.11	17.96	36.06	6.94	18.06	229.20
Ave Recip Rank	0.5022	0.7685	0.8416	0.4287	0.6360	0.7235	0.0046	0.6560	0.8242

Table 31. Retrieval performance across TREC-6 groups participating in the SDR track.

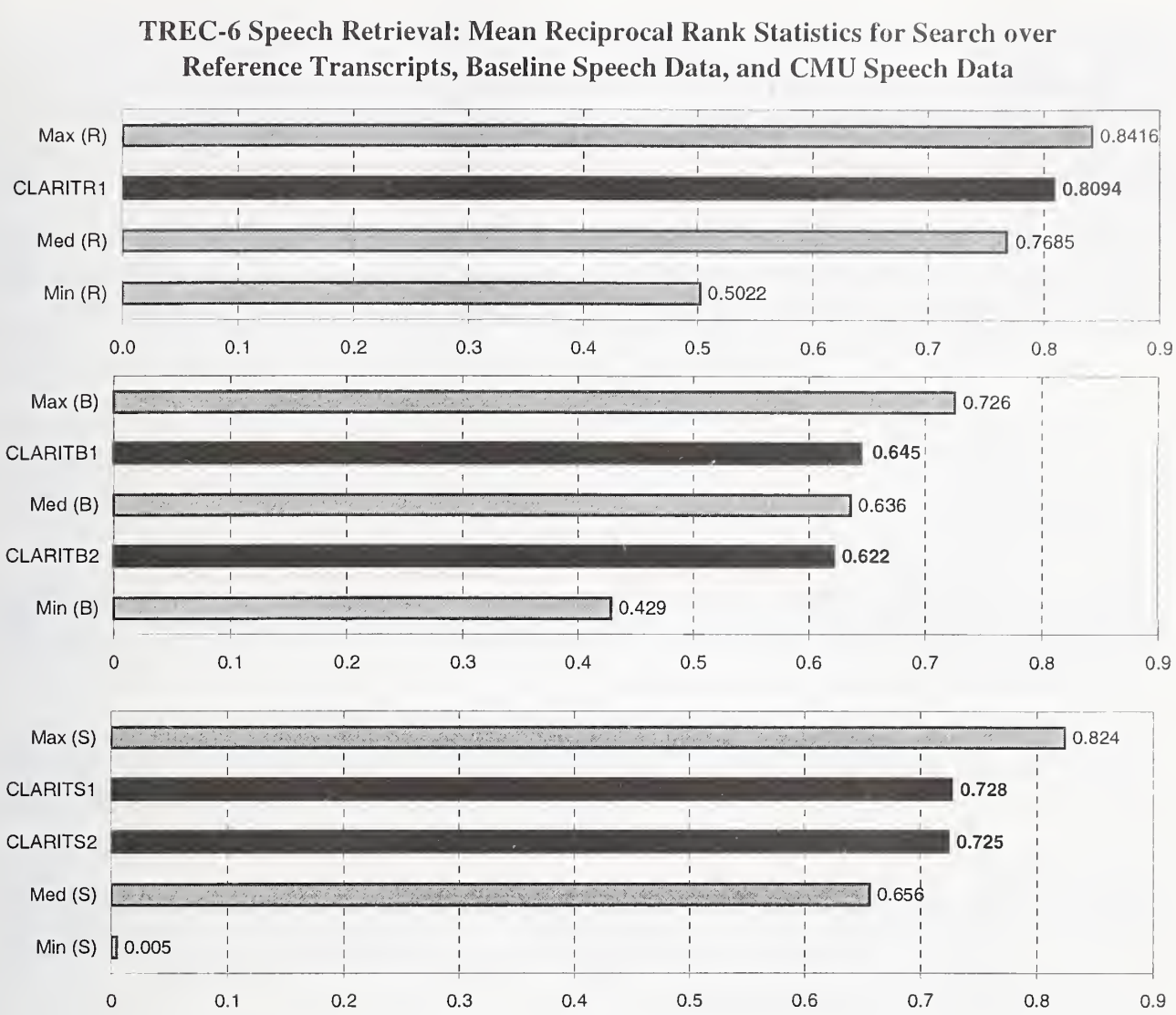


Figure 18. Comparison of the CLARIT average reciprocal rank with the group median.

Based on the percentage of items retrieved at the rank 1, the CLARIT experiment over the Informedia speech transcripts, CLARITS1, achieved the second best performance after the INQUERY Retrieval [Allan et al. 1998] system over the Dragon Systems speech transcripts. The same experiment was ranked third for performance over the reference data and fifth over the baseline speech data.

8 Conclusion

As we noted in the introduction to this paper, we believe that it is important to understand the effects of our techniques at the level of individual topics and to optimize our processing on a per-topic basis. We feel that we made significant progress toward this goal in our Routing, Filtering, and Ad-Hoc Retrieval experiments. In particular, our attempts to optimize training and term selection for individual topics have demonstrated both the potential value of such techniques as well as the difficulties in applying them consistently. Clearly, we will be focusing future efforts on this problem.

We are especially pleased with the strength of the CLARIT analysis (indexing), term-selection (query expansion), and matching (retrieval) modules. The system's performance in the Ad-Hoc Retrieval, Chinese, and SDR tracks is attributable, essentially, to these core processes. Indeed, in the case of Ad-Hoc Retrieval, we feel that the system may well be performing in a "natural" (and high) limit—gated principally by the variability of user-specific (subjective) judgments of relevance.

In general, we regard the challenges of the Filtering task, especially the problems of threshold setting, dynamic updating, and user modeling, as the most difficult and relevant issues for future research. We expect to devote much of our energy to these problems in our subsequent work.

References

[Allan et al. 1998] Allan, J., Callan, J., Croft, W.B., Ballesteros, L., Byrd, D., Swan, R. Xu, J., "INQUERY Does Battle With TREC-6". In Voorhees, E.M., and Harman, D.K. (Editors), *The Sixth Text Retrieval Conference (TREC-6)*. NIST Special Publication. Washington, DC: U.S. Government Printing Office, 1998.

[Buckley & Salton 1995] Buckley, Chris, and Salton, Gerard, "Optimization of relevance feedback weights". In Fox, Ed, Ingwersen, Peter, and Fidel, Raya (Editors), *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995, 351–357.

[Buckley et al. 1997] Buckley, Chris, Singhal, Amit, Mitra, Mandar, "Using Query Zoning and Correlation Within SMART: TREC 5". In Voorhees, E.M., and Harman, D.K. (Editors), *The Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238. Washington, DC: U.S. Government Printing Office, 1997, 105–118.

[Cormack et al. 1998] Cormack, G.V., Palmer, C.R., To, S.S.L., Clarke, C.L.A., "Passage-Based Refinement (MultiText Experiments for TREC-6) ". In Voorhees, E.M., and Harman, D.K. (Editors), *The Sixth Text Retrieval Conference (TREC-6)*. NIST Special Publication. Washington, DC: U.S. Government Printing Office, 1998.

[Evans 1990] Evans, David A., "Concept Management in Text via Natural-Language Processing: The CLARIT Approach". *Working Notes of the 1990 AAAI Symposium on "Text-Based Intelligent Systems"*, Stanford University, March, 27–29, 1990, 93–95.

[Evans et al. 1991] Evans, David A., Ginther-Webster, Kimberly, Hart, Mary, Lefferts, Robert G., Monarch, Ira A., "Automatic Indexing Using Selective NLP and First-Order Thesauri". In A. Lichnerowicz (Editor), *Intelligent Text and Image Handling. Proceedings of a Conference, RIAO '91*. Amsterdam, NL: Elsevier, 1991, 624–644.

[Evans et al. 1993] Evans, David A., Lefferts, Robert G., Grefenstette, Gregory, Handerson, Steven K., Hersh, William R., Archbold, Armar A., "CLARIT TREC Design, Experiments, and Results". In Harman, Donna K. (Editor), *The First Text REtrieval Conference (TREC-1)*. NIST Special Publication 500-207. Washington, DC: U.S. Government Printing Office, 1993, 251–286; 494–501.

[Evans et al. 1996] Evans, David A., Milic-Frayling, Natasa, Lefferts, Robert G. "CLARIT TREC-4 Experiments". In Harman, Donna K. (Editor), *The Fourth Text REtrieval Conference (TREC-4)*. NIST Special Publication 500-236. Washington, DC: U.S. Government Printing Office, 1996, 305–321.

[Evans & Lefferts 1994] Evans, David A., and Lefferts, Robert G., "Design and Evaluation of the CLARIT–TREC-2 System". In Harman, Donna K. (Editor), *The Second Text REtrieval Conference (TREC-2)*. NIST Special Publication 500-215. Washington, DC: U.S. Government Printing Office, 1994, 137–150.

[Evans & Lefferts 1995] Evans, David A., and Lefferts, Robert G., "CLARIT–TREC Experiments". *Information Processing and Management*, Vol. 31, No. 3, 1995, 385–395.

[Evans & Zhai 1996] Evans, David A., and Zhai, Chengxiang, "Noun-Phrase Analysis in Unrestricted Text for Information Retrieval". *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, Santa Cruz, California, June, 1996. San Francisco, CA: Morgan Kaufmann Publishers for the Association for Computational Linguistics, 1996, 17–24.

[Hosmer & Lemeshow 1989] Hosmer, David W. and Lemeshow, Stanley, *Applied Logistic Regression*, Wiley Series in Probability and Mathematical Statistics (Shewhart and Wilks, eds), Wiley and Sons, New York etc, 1989.

[McCullagh & Nelder 1989] McCullagh, Peter J. and Nelder, John A., *Generalized Linear Models* (2nd edition), Monographs in Statistics and Applied Probability (Cox, Hinkley, Rubin and Silverman eds.), Chapman & Hall, 1989.

[Milic-Frayling et al. 1996] Milic-Frayling, Natasa, Zhai, Chengxiang, Tong, Xiang, Mastroianni, Michael, Evans, David A., Lefferts, Robert G., "CLARIT TREC-4 Interactive Experiments". In Harman, Donna K. (Editor), *The Fourth Text REtrieval Conference (TREC-4)*. NIST Special Publication 500-236. Washington, DC: U.S. Government Printing Office, 1996, 323–357

[Milic-Frayling et al. 1997] Milic-Frayling, Natasa, Tong, Xiang, Zhai, Chengxiang, Evans, David A., "CLARIT Compound Queries and Constraint-Controlled Feedback in TREC-5 Ad-Hoc Experiments". In Voorhees, E.M., and Harman, D.K. (Editors), *The Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238. Washington, DC: U.S. Government Printing Office, 1997, 315–334.

[Robertson et al. 1997] Robertson, Stephen E., Beaulieu, M.M., Gatford, M., Huang, Xiangji, Walker, S., Williams, P., "Okapi at TREC-5". In Voorhees, E.M., and Harman, D.K. (Editors), *The Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238. Washington, DC: U.S. Government Printing Office, 1997, 143–165.

[Robertson & Sparck-Jones 1976] Robertson, Stephen E., and Sparck-Jones, Karen, "Relevance weighting of search terms". *Journal of the American Society Information Science*, 27, 1976, 129–146.

[Siegler et al. 1998] Siegler, M.A., Slattery, S.T., Seymore, K., Jones, R.E., Hauptmann, A.G., Witbrock, M.J., "Experiments in Spoken Document Retrieval at CMU". In Voorhees, E.M., and Harman, D.K. (Editors), *The Sixth Text Retrieval Conference (TREC-6)*. NIST Special Publication. Washington, DC: U.S. Government Printing Office, 1998.

[Singhal 1998] Singhal, Amit, *"AT&T at TREC-6"*. In Voorhees, E.M., and Harman, D.K. (Editors), *The Sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication. Washington, DC: U.S. Government Printing Office, 1998.

[Schutze et al. 1995] Schutze, Hinrich, Hull, David A., and Pederson, Jan O., "A comparison of classifiers and document representations for the routing problem". In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995, 229–237.

[Tong & Evans 1996] Tong, Xiang and Evans, David A., "A Statistical Approach to Automatic OCR Error Correction in Context". *Proceedings of the Fourth Workshop on Very Large Corpora (WVLC-4)*, Copenhagen, Denmark, August 4, 1996, 88–100.

[Tong et al. 1997a] Tong, Xiang, Zhai, Chengxiang, Milic-Frayling, Natasa, Evans, David A., "Experiments on Chinese Text Indexing—CLARIT TREC-5 Chinese Report". In Voorhees, E.M., and Harman, D.K. (Editors), *The Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238. Washington, DC: U.S. Government Printing Office, 1997, 335–339.

[Tong et al. 1997b] Tong, Xiang, Zhai, Chengxiang, Milic-Frayling, Natasa, Evans, David A., "OCR Correction and Query Expansion for Retrieval on OCR Data—CLARIT TREC-5 Confusion Track Report". In. Voorhees, E.M., and Harman, D.K. (Editors), *The Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238. Washington, DC: U.S. Government Printing Office, 1997, 341–345.

[Voorhees et al. 1997] Voorhees, Ellen M., and Harman, Donna K. (Editors), *The Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238. Washington, DC: U.S. Government Printing Office, 1997.

[Walker et al. 1998] Walker, S., Robertson, S.E., Boughanem, M., Jones, G.J.F., Sparck Jones, K. "Okapi at TREC-6. Automatic ad hoc, VLC, routing, filtering and QSDR". In Voorhees, E.M., and Harman, D.K. (Editors), *The Sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication. Washington, DC: U.S. Government Printing Office, 1998.

[Zhai et al. 1997] Zhai, Chengxiang, Tong, Xiang, Milic-Frayling, Natasa, Evans, David A., "Evaluation of Syntactic Phrase Indexing—CLARIT TREC-5 NLP Track Report". In Voorhees, E.M., and Harman, D.K. (Editors), *The Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238. Washington, DC: U.S. Government Printing Office, 1997, 347–357.

CSIRO Routing and Ad-Hoc Experiments at TREC6

Arkadi Kosmynin

Research Data Network Co-operative Research Centre
& CSIRO Mathematical and Information Sciences
723 Swanston St, Carlton, Victoria 3053, Australia

Arkadi.Kosmynin@vic.cmis.csiro.au

Background

CSIRO stands for Commonwealth Scientific and Industrial Research Organization. It is the Australian Government's main research body. This is the first year CSIRO is taking part in TREC. We got involved in textual information retrieval research as a part of our activities in Resource Discovery Unit at the Research Data Network Co-operative Research Centre. The primary aim of our research in IR is improvement of the efficiency of resource discovery systems and networked information retrieval.

General Discussion

The classic vector space model [1] has served very well for the purpose of textual information retrieval. But text is much more than just a set of terms. Firstly, the actual meaning of a word depends on the context in which the word is appearing. Secondly the same words can be combined in different ways to produce texts of different meaning. The classic model takes advantage of redundancy which exists in texts, but how far can we go exploiting this redundancy?

In our view, the main source of improvement of IR efficiency is not in taking similarity measures to perfection, but in using additional information from the text. Good examples of such information are context in which words appear in texts, words order and proximity.

The idea to use this information is not a new one. There is much of research directed at taking advantage of the additional information available in texts. There is also strong evidence that documents can not be treated as homogeneous objects [2]. The ideas of Local Context Analysis (LCA) [3], passage retrieval [2,4] and word sense disambiguation, - all suggest that we are on the right track.

Our own experiments on extended use of context [5] conducted on the Reuters-22173 collection illustrate that use of context can be of a great value, as we achieved very good results using this method. Our first goal in TREC was to test this method on a different collection. To our disappointment, it did not work at all, though, after some consideration we found a possible reason. The main difference between the two collections is that in the Reuters collection all articles are relatively small, and in the TREC collection many articles are large. This means that the assumption that the documents are monocontextual holds more or less for the Reuters collection, but is not true for the TREC collection. Unfortunately, our method depended on this assumption. It was clear that in TREC we had to use proximity information to define context, and we had to devise a method to do that.

The method we finally developed is entirely different from the one described in [5]. It is similar to LCA and passage retrieval, but takes these ideas further. The main idea is that if we have a query and are looking for documents which are the best answers to the query, then 1) the context in which the query words occur in the documents under consideration does matter, and 2) we are only interested in parts of the document which are relevant to this query, the total length of the document is irrelevant (unless we are concerned with the cost of retrieval of very long documents).

In our method, we do not compare a query to a document. We compare context of the query to the context in which query words appear in the document. Comparison of two similar objects (namely, two contexts) seems to be a more reasonable thing to do than to compare two different objects - like a query and a document. Besides, this method seems to model exactly what human beings do when asked a question. If context of the question is unclear, we ask for elaboration; when we believe we have got the context right, we are trying to identify information which has something to do with the context, not necessarily with the exact words of the question. Even if we are asked for very particular information (e.g. a description of event happened on a particular date), we first try to find contextually close information and then filter it to identify items dealing with the particular details we have been asked about.

To perform the task of comparison in our method, we first expand the query to its context in the top relevant documents, exactly as in LCA. Then we reduce every document to be compared to the context of the query (words surrounding the original query words in the document). Then we compare the two contexts.

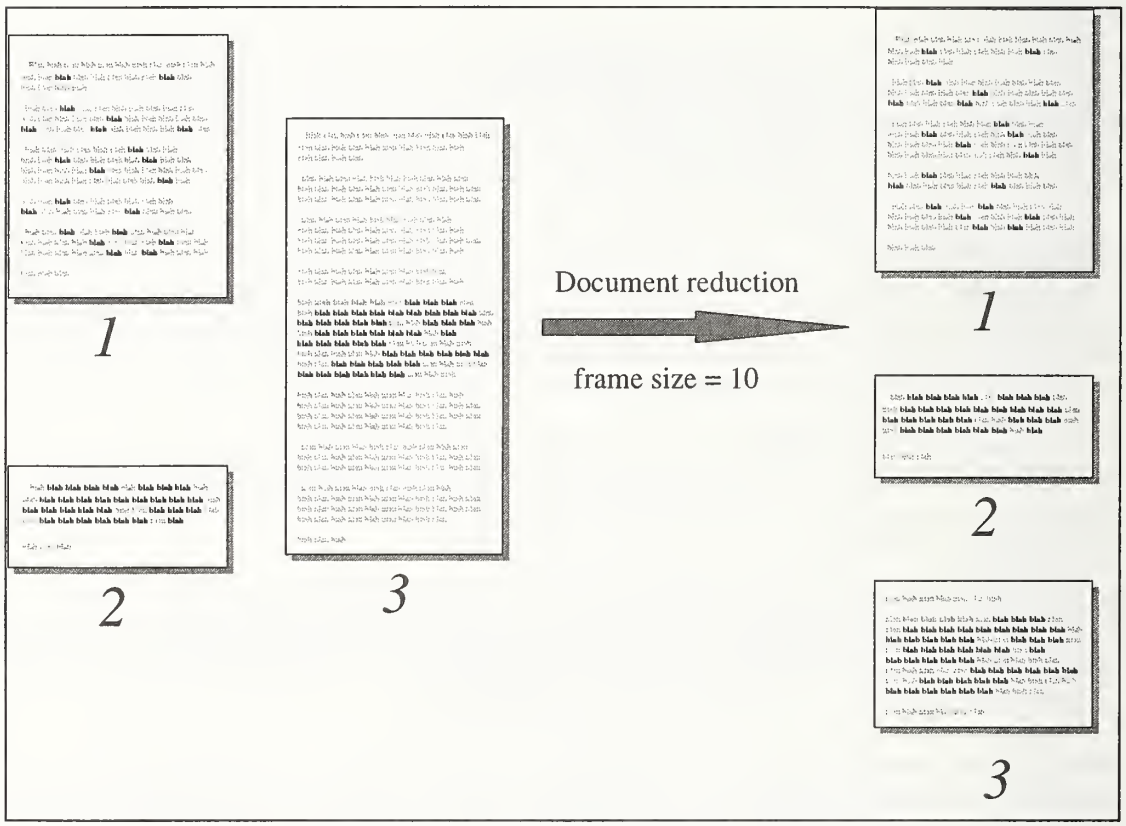


Figure 1. Example of document reduction

The process of reduction is illustrated in Figure 1. Bold parts of text are query terms. Document 1 is irrelevant to the query, but contains accidental query terms. Document 2 is relevant and short. Document 3 contains a relevant part, but is considerably longer than the other two and therefore would be assigned the lowest score of these 3 documents. However, the 3rd document will not be penalized for its length after document reduction. It may well be the case that the documents would be ranked in order 2,3,1. Document reduction facilitates better scoring and provides refined data for training.

We learned that behavior of "document context vectors" (as we call documents reduced to context of a particular query) seems to be quite different from the reported behavior of document vectors. For example, there are only very few of the top positive and negative examples needed to train a classifier. Training on anything but the top scoring documents (both relevant and irrelevant) does not produce any improvement. Negative weights do play their role well. We attribute these differences to the fact that we are dealing with refined material. (At this stage, however, we can not be completely sure of this because the results on the

test set were unexpectedly low. A further investigation is required.) By reducing documents to context vectors, we are removing a lot of irrelevant information, and hope not to be removing too much of relevant information. Thus, influence of accidental dependencies is significantly reduced. On the other hand, the core dependencies are enforced.

There is one more advantage of this method which should be mentioned. As it can be seen below in the description of context vectors construction, this method favors documents where the query terms appear close together. For example, it will give higher scores to long documents with a couple of paragraphs containing concentrated query words, than to comparatively short documents where query terms are widely spread - something that the classic vector space model is unable to do.

Document Context Vectors

To construct context vector of a <document, query> pair, we take words of the document appearing within a given range (we call this range "frame size" - FS) of the query terms in the document. The query terms in the document are also included. Note that if occurrence of the query terms in the document is accidental, they will tend to be spread and surrounded by random terms. This will result in a relatively long vector consisting of random terms with low weights. If the document is relevant, the query terms will tend to appear together and surrounded by similar terms (context terms). Frames of the query terms will overlap. It will result in a shorter vector consisting of (context) terms with higher weights.

As it can be seen, we are getting an advantage from two sources. The first source is the consideration of context terms. The second source is penalizing documents where query terms are spread. As we have learnt, these sources have different dependence on the FS parameter. With higher values of FS, more accidental terms are included in the context vectors. It makes context "polluted" and reduces value of the context component. On the other hand, greater value is received from overlapping frames if query terms occur in a close group. This increases value of the proximity component.

Construction of document context vectors is very cheap, especially for systems that support indexing of proximity information.

Query Context Vectors

To construct query context vectors, we use the Rocchio [6] algorithm, but we run it on document context vectors, instead of source documents. We order the resulting vectors by the weights of the words and truncate them to a given number of words (words in vector - WV).

Routing Experiments

Training

To train a classifier for a query, we first ordered all documents according to their relevance scores (top scores first) obtained using the MG query engine [7]. We then used top Npos positive and top Nneg negative examples to train the classifier. We optimized parameter settings using data sets from the previous TREC conferences.

Routing

In the routing stage, all the query vectors were placed in RAM. Every document was read once, a document context vector was created for every query and comparison performed. The best 1000 scores together with document's ids were kept for each query and printed out when all the documents had been processed.

The system was re-trained with new scoring information available and another iteration was performed.

Parameter Settings

Run N1 (automatic):

Iteration 0: MG used to obtain initial scores.
Iteration 1, training: FS = 3, WV = 400, Npos = 10, Nneg = 20, $\alpha = 0$, $\beta = 10$, $\gamma = 4$;
Routing: FS = 3.
Iteration 2, training: FS = 3, WV = 400, Npos = 10, Nneg = 20, $\alpha = 0$, $\beta = 10$, $\gamma = 4$;
Routing: FS = 3.

Run N2 (automatic):
Iteration 0: MG used to obtain initial scores.
Iteration 1, training: FS = 3, WV = 400, Npos = 10, Nneg = 25, $\alpha = 0$, $\beta = 16$, $\gamma = 4$;
Routing: FS = 3.
Iteration 2, training: FS = 3, WV = 400, Npos = 10, Nneg = 25, $\alpha = 0$, $\beta = 16$, $\gamma = 4$;
Routing: FS = 3.

(α, β, γ are the parameters of the Rocchio algorithm.)

Cost

We used a PC with 96Mb of RAM and Pentium II 266MHz processor for these experiments.

Dictionary creation took 20 minutes, training took 1 second per query (on 35 examples), routing run took 0.5 hour per iteration. These times can be significantly reduced in a system that retains proximity information in indexing stage. As our software looses this information, we had to re-scan every document to create document context vector for a query.

Results

Run	at 5 docs	at 10 docs	at 20 docs	at 200 docs	R-precision	Average
N1	0.4894	0.4468	0.3957	0.2080	0.2552	0.2068
N2	0.4936	0.4468	0.3872	0.2078	0.2603	0.2053

Ad-Hoc Experiments

The major difference between routing and ad-hoc query processing is that there is no training data for ad-hoc queries. We have to rely on assumption that the first few top scoring documents are mostly relevant to the query. Given this assumption, we can use these documents as positive examples for building query context vectors, with no negative examples. The rest of the process is identical to routing.

Query Context Vectors Construction

To obtain initial scoring (iteration 0), we assume that the query context vector is the original query. We build a document context vector for every document and obtain a score by comparing it to the query context vector. This process gives slightly better results than the classic vector space model (MG) because it penalizes documents where query terms are widely spread.

We use the top scoring documents and the Rocchio algorithm to construct query context vectors.

Query Processing

The query processing stage is identical to the routing stage in the description of routing experiments above.

Parameter Settings

Run N1 (automatic, full queries):
Iteration 0: query context vector = query, FS = 3;
Iteration 1, training: FS = 10, WV = 400, Npos = 5, Nneg = 0, $\alpha = 0$, $\beta = 1$, $\gamma = 0$;
Processing: FS = 10;
Iteration 2, training: FS = 10, WV = 400, Npos = 6, Nneg = 0, $\alpha = 0$, $\beta = 1$, $\gamma = 0$;
Processing: FS = 10;

Run N2 (automatic, description-only queries):

Iteration 0: query context vector = query, FS = 3.

Iteration 1, training: FS = 17, WV = 400, Npos = 4, Nneg = 0, $\alpha = 0$, $\beta = 1$, $\gamma = 0$;

Processing: FS = 17.

Run N3 (automatic, title-only queries):

Iteration 0: query context vector = query, FS = 3.

Iteration 1, training: FS = 17, WV = 400, Npos = 4, Nneg = 0, $\alpha = 0$, $\beta = 1$, $\gamma = 0$;

Processing: FS = 17.

Cost

We used a PC with 96Mb of RAM and Pentium II 266MHz processor for these experiments.

Dictionary creation took 1.5 hour, training took less than 1 second per query, processing run took 2.75 hour per iteration. These times can be significantly reduced in a system that retains proximity information in indexes.

Results

Run	at 5 docs	at 10 docs	at 20 docs	at 200 docs	R-precision	Average	MG Aver.
N1	0.3360	0.2820	0.2180	0.0756	0.1455	0.1265	0.1208
N2	0.2720	0.2280	0.1840	0.0584	0.1295	0.1171	0.0904
N3	0.3440	0.2860	0.2290	0.0760	0.1481	0.1259	0.1207

Fault Analysis and Conclusion

This method is simple and practical. It demonstrated good performance on the data set which we used for preliminary experiments. However, the routing results on the test data set are disappointing (50% lower in precision). We initially blamed overfitting for this, but later experiments have shown that overfitting is not the cause. Even if we train the system on the TREC-6 routing test set and then run it on the same set, we are still getting significantly lower results than on our training set.

Our analysis has shown that we could improve performance if we had trained the system on all available data, not just on FBIS documents, - something that was hard to do because of our hardware limitations at that time.

We have conducted more ad-hoc experiments after the conference and obtained much better results. We are investigating whether this improvement was a result of different parameters or removing some minor bugs from the software. We intend to publish this work and results later.

We could improve performance if we applied proven techniques commonly used by other groups, but it was beyond the scope of the first year of participation. We feel that we have accomplished our goals for the first year and there is a large amount of work to do ahead.

Acknowledgments

The work reported in this paper has been funded in part by the Co-operative Research Centres Program through the Department of the Prime Minister and Cabinet of the Commonwealth Government of Australia.

References

[1] G. Salton. Automatic text processing: the transformation, analysis and retrieval of information by computer. Addison-Wesley, 1989, 387 p., ISBN 0-201-12227-8.

[2] R. Wilkinson. Effective retrieval of structured documents. In Proceedings, SIGIR 94, pp. 311-317, Dublin, Ireland, 1994. Association for Computing Machinery.

[3] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In Proceedings, SIGIR 96, pp. 4-11, Zurich, 1996. Association for Computing Machinery.

[4] G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in information systems. In Proceedings, SIGIR 93, pp. 49-58, New York, 1993. Association for Computing Machinery.

[5] A. Kosmynin and I. Davidson. Using background contextual knowledge for documents representation. In Principles of Document Processing '96 Workshop, California, 1996.

[6] Jr. J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313-323, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[7] I. H. Witten, A. Moffat, and T. C. Bell. Managing Gigabytes, Van Nostrand Reinhold, New York, 1994.

Ad hoc Retrieval Using Thresholds, WSTs for French Mono-lingual Retrieval, Document-at-a-Glance for High Precision and Triphone Windows for Spoken Documents

Alan F. Smeaton[1], Fergus Kelledy[2] and Gerard Quinn[1]

[1]School of Computer Applications
Dublin City University
Glasnevin, Dublin 9, IRELAND.

[2]Broadcom Éireann Research
Dublin, IRELAND

Abstract:

This paper describes work done by a team from Dublin City University as part of TREC-6. In this TREC exercise we completed series of runs in 4 categories. The first was the mainline ad hoc retrieval task in which we repeated our entry for TREC-5, without modification. This is based on applying various thresholds to processing a query including query term and posting list thresholds, in order to improve retrieval efficiency. As our previous work has shown, this can be done without any loss in retrieval effectiveness. Our second set of submitted runs were as part of the cross-lingual retrieval track where we ran French topics against French texts, effectively mono-lingual retrieval. What is novel about our approach is that it is based upon matching word shape tokens derived from character shape codes, rather than matching word stems or base forms. This technique is useful for retrieving from scanned document images rather than full texts and is something we are currently refining for English texts (and English queries). With those other experiments we have obtained surprisingly effective retrieval and this venture in TREC-6 was to see how effective WST-based retrieval could be for French. The third series of experiments we submitted were based on the high precision track in which we used a graphical representation of a ranked list of documents and the positional occurrences of search terms within those top-ranked documents, relative to each other. Our final experiments were as part of the spoken document retrieval track in which we removed the tags used for story bounds, turned transcripts and topics into a phonetic representation using a phoneme dictionary and we then retrieved story identifiers based on a triphone match between topic and fixed-width windows of triphones in the transcripts. We also applied a weighting function to triphones as they occurred in story "windows" based on their offset within those windows.

1. Introduction

TREC-6 is Dublin City University's fourth consecutive year for involvement in TREC and our largest to date. Our work is neatly divided into four distinct areas representing the mainline ad hoc retrieval task and three of the specialist tracks. Each is described here in turn and conclusions are drawn about each of our work areas.

2. Mainline Ad Hoc Retrieval Task

Our submissions to the mainline ad hoc retrieval task in TREC-6 were the same as our submissions for TREC-5 and in fact we used the exact same parameter settings as in TREC-5. This work is based on reducing the search space for processing a query, and hence the execution time, by applying a combination of thresholds to the processing. These include reducing the number of search terms looked up in the inverted file, the proportion of entries in the inverted file for a given search term, and the total number of document accumulators or registers used to hold document scores, i.e. the total number of documents assigned a score of any kind. While the motivation for this work is to reduce the processing time we have repeatedly shown in TREC-5 and in experiments on the data from TREC-4 and TREC-3 that this can be accomplished without loss of retrieval effectiveness, and indeed with a marginal improvement in precision and recall figures. We refer the reader to other publications [Smeaton & Kelledy, 1997a and 1997b] for further details of the methods we use.

The results we obtained for our TREC-6 submissions are summarised below in Table 1. We present the results of two runs, both based on using the full topic description but the first run (**DCU97lnt**) used no thresholding while the second run (**DCU97lt**) used the query term, postings list and accumulator threshold settings used in TREC-5. These TREC-5 settings were used blind in TREC-5 and were based on runs on the TREC-3 and TREC-4 data set. The settings we used in TREC-6 are similarly blind, i.e. they remain untuned to the data set. Comparing these two results we see that the thresholding has an enormous improvement on retrieval effectiveness, probably because of the number of noise words from the full topic description that are discarded by the query term thresholding.

The results for the other two runs we submitted in the ad hoc retrieval task are also shown in Table 1 and are based on using the short version of the topics (description only) in run **DCU97snt** and based on using the topic titles only in **DCU97vs**. Both of these latter runs are done with no thresholding used. In later work to be reported elsewhere we shall explore how applying our best thresholding settings impacts retrieval effectiveness and execution speed on these runs.

	DCU97lnt	DC97lt	DCU97snt	DCU97vs
	Full topic, no thresholding	Full topic, thresholding	Description only, no thresholding	Title only, no thresholding
No. relevant documents found	1488	1637	1796	2180
P @ 0.0	0.2384	0.3388	0.5179	0.6016
0.1	0.1143	0.1895	0.2917	0.4086
0.2	0.0646	0.1362	0.2374	0.3343
0.3	0.0513	0.0930	0.1542	0.2570
0.4	0.0391	0.0770	0.1334	0.2232
0.5	0.0222	0.0448	0.1099	0.1830
0.6	0.0163	0.0325	0.0843	0.1421
0.7	0.0095	0.0185	0.0655	0.0988
0.8	0.0053	0.0085	0.0483	0.0554
0.9	0.0016	0.0015	0.0144	0.0341
1.0	0.0002	0.0009	0.0121	0.0325
Avg. Precision	**0.0372**	**0.0696**	**0.1296**	**0.1941**
P @ 5 docs	0.0960	0.1600	0.2800	0.3800
P @ 10 docs	0.0820	0.1460	0.2000	0.3280
P @ 30 docs	0.0687	0.1127	0.1500	0.2553

Table 1: Results for mainline ad hoc retrieval task

3. Character Shape Coding for Mono-Lingual French Retrieval

Character shape codes (CSCs) are a reduced alphabet into which the full range of case-sensitive alphanumeric characters that can occur in printed form in documents, can be mapped. This mapping is based on image processing considerations relative to the task of identifying characters using a kind of OCR process. Essentially, characters with similar "shapes" are mapped into the one CSC and words in the original document are mapped into word shape tokens (WSTs).

Character shape codes were initially developed as a means to identify the language being used in a document. Since then they have been used as a pre-process for full-scale optical character recognition [Spitz, 1997], and for word-spotting from images [Spitz, 1995]. In TREC-5 we attempted to use character shape codes and WSTs for information retrieval, simulating the situation where printed documents would have been scanned and CSC recognition applied to the images instead of full-scale OCR. This work has proceeded with further experiments reported in [Smeaton & Spitz, 1997] showing exciting promise for the technique. The work reported here is an attempt to apply WST-based indexing to French documents. Document texts are turned into their WST equivalents and indexed by these WSTs. Topics are also encoded as WSTs and retrieval is based on matching WSTs instead of word stems.

Several scenarios could motivate CSC recognition including poor quality original documents (Faxes, photocopies of photocopies, etc.) and the need to reduce computing overheads as CSC recognition is an order of magnitude faster than OCR and much more accurate. A number of CSC mappings have been defined but in this work we use the simplest, and thus the fastest and most accurate, known as V_0. To illustrate the mapping, in figure 1 below we have drawn the baseline and the x-line (top of the lowercase character x) on a sample of text.

Figure 1: Sample of Word Shape Tokens

Using a few simple rules we can devise a mapping based on whether each character rises above the x-line [A-Zbdfhklt], dips below the baseline [gpqy], does neither [acemnorsuvwxz], does neither but has a dot above it [i] or dips below and has a dot above it [j]. Using the representative letters **A, g, x, i, j** we can represent the text in Figure 1 as Aaix ix xg jxA. To adapt this mapping to the character set used in French, any lowercase accented characters such as é, ì or ô are mapped to the CSC "i" while ç is mapped to a "g". Uppercase accented characters are mapped to "A". Punctuation characters are discarded.

While aggregating similar shaped letters into one does yield a huge loss of information, for English we have found the uniqueness of some WSTs to be surprising, enough on which to develop an information retrieval system based on WST matching and that is what we try here for French.

One of the major difficulties with using WSTs for information retrieval is that each document is represented by the surface form of the word occurrence and variations in surface form (case mixing) and in word morphology (plurals, word endings) must be taken into account at the query processing stage. Thus when a user inputs a search term, all legitimate variations of that term due to case changes and word morphology must be generated at query time and used as search terms. In TREC-5, working on English texts, we found that approach to generate many search terms whose surface form is shared by many other surface forms and this is effectively adding many noise terms to the query. As an extreme example, the word forms *Pommes, terres, Mesure* all generate the WST Axxxxx as do 1,285 **other** surface forms we came across while the surface forms *terre* and *luxes* generate Axxxx which is shared by 1412 surface forms in total that we know of. To compensate however, sometimes the mapping can be surprisingly unique; the words *religion, autrichian, recyclage* and *automobiles* all have unique WSTs in the texts we processed

In order to generate the morphological variants of words in the topics, the topics were linguistically analysed by the Xerox processor at the Rank Xerox Research Laboratory in Grenoble and for the base form of each words in the topic descriptions, morphological variations were generated. These were then post-processed to generate surface variations such as starting capital letters if they could have occurred as the first word in a sentence, and then these terms were turned into their WST form thus creating our topics. As a sample text corpus from which to generate frequency statistics, approximately two-thirds of the document texts were processed to record each unique surface form occurrence and the number of times that form occurred in the sample texts. This yielded 128,380 unique surface forms, a good deal short of the c.300,000 we have for English texts but sufficient to

work with. Many of these surface form occurrences would be spelling errors but we did not clean this list in any way. Content-bearing terms were manually identified in each of the topics and all the surface forms associated with those content-bearing terms were concatenated to constitute our starting point for our topics.

Rather than simply turn these word lists into WSTs and run them against the documents, which we realised from TREC-5 experiments would not be effective, we submitted two runs for assessment. In the first run (DCU97Fv1, which was not judged) we used only the WSTs derived from topic tokens where the WST was shared by no more than 10 other surface forms in the list of 128,380 forms we had recorded. In some post-TREC-5 experiments on English texts we found this to be crude but surprisingly effective [Smeaton & Spitz, 1997]. In our second run (DCU97Fv2 which was judged) we manually selected WSTs for each topic based not only on the number of surface forms sharing a WST but on the importance of that WST to the topic. For example, topic CL9 (not our best-performing topic) is entitled "Les effets de déforestation" and the tokens (and the total number of other word tokens sharing that WST) are:

effet (3) effets (1)	Effet (18)	Effets (10)	déforestation (1)
déforestations (1)	désertification (1)	désertifications (0)	frêner (177)
changement (4)	changements (3)	climat (5)	Climat (8)
épuisement (1)	épuisements (0)	terre (1412)	terres (1288)
inondation (3)	inondations (1)	ouragan (41)	ouragans (25)

Clearly words like *terre* and *frêner* should be eliminated and pruning such terms based on WST frequency would not hurt this query too badly. This would have corresponded to the tokens used in run DCU97Fv1. For our second run, DCU97Fv2, we would have judiciously omitted some terms but left others in, using the number of shared WSTs per search term as an **indicator** rather than the deciding factor on whether to include a search WST or not. For example, the terms *changement* and *changements* would have been excluded even though the uniqueness of their WSTs suggest they be left in. In fact, for this query, such adjustments did improve the performance of the topic. Tokens with zero above obviously did not appear in the sample of document texts we used to record possible WSTs.

To illustrate a case where simple pruning based on frequency does harm a query, if we look at topic CL11 entitled "Le coton écologique" we see the following terms and their frequencies:

coton (135)	Coton (616)	écologique (2)	production (10)	usage (34)
usages (64)	bénéfice (19)	bénéfices (10)	positif (2)	positifs (1)
positive (5)	positives (6)	terre (1412)		

Eliminating terms with frequency >10 leaves no content-bearing terms at all and indeed this topic retrieves no relevant documents at all for us in the top-1000 out of the 8 known relevant topics in the corpus.

For the record, the number of WST terms included in the topics for the v1 run (automatic pruning based on frequency) was 10.0 on average while the manually chosen WSTs in the v2 run yielded

an average of 7.8 terms per topic. The original set of surface form occurrences chosen manually as content-bearing yielded 17.8 surface forms per topic. In terms of performance, the precision-recall figures for our runs using the 21 judged topics, are given below in Table 2.

	DCU97Fv1 (automatic terms)	**DCU97Fv2 (manual choice)**
R @ 0.00	0.3833	0.3580
0.10	0.2374	0.2471
0.20	0.1982	0.1940
0.30	0.1497	0.1657
0.40	0.1238	0.1322
0.50	0.1029	0.1067
0.60	0.0673	0.0702
0.70	0.0607	0.0600
0.80	0.0413	0.0388
0.90	0.0366	0.0377
1.0	0.0085	0.0087
Avg P	**0.1073**	**0.1102**
P @ 5 docs	0.1905	0.1905
P @ 10 docs	0.2143	0.2095
P @ 30 docs	0.1698	0.1730

Table 2: Performance of WST-based French mono-lingual retrieval

These results are poor in terms of precision-recall, especially averaged over the whole range, though higher precision is what we would have hoped for in terms of overall performance. The reader is reminded that we are simulating a retrieval situation in which scanned images of documents are retrieved based on the shapes of words occurring within them, albeit by a 100% accurate CSC recognition process.

In terms of comparison to submitted runs by other groups we have one topic which is the median, two topics (three in the case of V1) which are above median and the remainder of the 21 are below the median (as measured by average precision. However, WST-based retrieval works reasonably well for some topics, but terribly badly for others, even with the manual selection of WST search terms. The graph in Figure 2 shows the average Precision for each of the judged topics for each of our runs and also the average Precision per topic for all the submitted runs by all participating groups. This average Precision across all submitted runs gives an indication of the degree of difficulty of each topic. Figure 2 clearly shows that for some topics we perform reasonably well compared to others (topics 1, 6, 9, 12, 16 and 20) but there are others where we perform badly, even with manually selected WST search terms, and where the topic is not so difficult (topics 5, 7, 8, 10, 13, 14, 17, 18, 19, 21, 23 and 24) and it is the performance of these latter topics that brings down our overall performance figures.

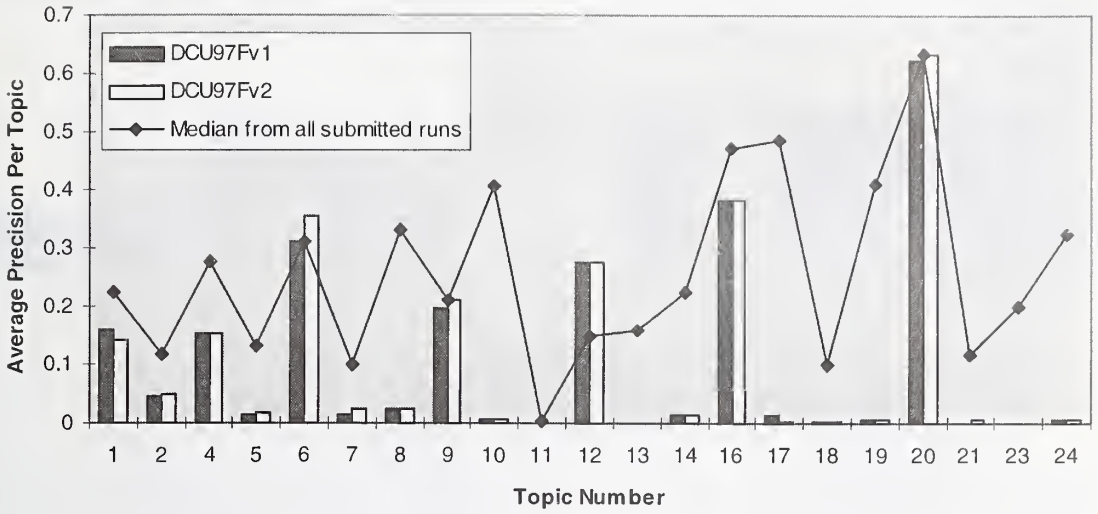


Figure 2: Avg Precision per topic for DCU runs and Median from all submitted runs

What is somewhat surprising in our results is the fact that v2 (manual pruning of search terms) is not considerably better than v1 (automatic pruning). We would have expected that manually choosing search terms based on WST frequencies and on importance to the query would have been much more effective than doing so by using an automatic frequency threshold, though perhaps this may be due to the small sample size (of topics evaluated).

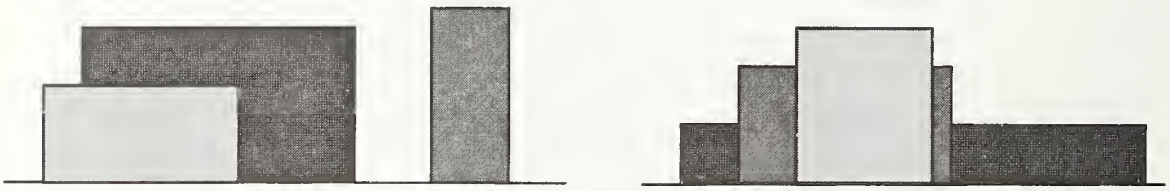
We are encouraged to continue with this work, though for English rather than for French and the progression of this work will be reported elsewhere.

4. "Document-at-a-Glance" for High Precision Retrieval

Several approaches have been developed within the last few years for presenting the results of a ranked retrieval of documents graphically. These have included techniques based on sophisticated visualisation such as LyberWorld [Hemmje et al, 1994] and Cat-a-Cones [Hearst & Pedersen, 1997] as well as those based on more simple graphical icons like Tilebars [Hearst, 1995] and the work by Veerasamy [Veerasamy & Heikes, 1997]. At Dublin City University we have developed a graphical iconic representation which has similar features to Tilebars and the work by Veerasamy. We call our representation the "Document-at-a-Glance" (DaaG).

A DaaG is an icon where the horizontal axis corresponds to the length of the document. Marked on this horizontal axis are the location of the first and the location of the last occurrence of each of the search terms which occur within that document. Using these "first" and "last" offsets, a rectangle can be drawn whose "height" is some function of the number of times that search term has occurred in the document, and the IDF weight of that search term. Clearly, the higher this "box", te more important in terms of weight and/or number of occurrences that search term is to that document. Such rectangles are drawn for each search term that occurs in each of the top-ranked documents and these are laid out in such a way that all are visible. The boxes are coloured and the colours match the colours assigned by the system to each of the search terms in the topic.

For example, consider the query "computer science education" and the hypothetical DaaGs generated for two documents as shown below.



where the "colour" codes[1] correspond to computer science education

If we assume that the RSVs for the two documents are about the same then clearly the document on the left is about computer education but not about computer science whereas the document on the right is more likely to be about computer science education because those three terms co-occur in the same parts of the document. With the Document-at-a-Glance icon, a user can have such a summary of search term occurrences generated visually. In our implementation we generate DaaGs for all top-ranked documents and let the user see these before deciding whether a document text is worth the effort of looking at. Double-clicking on a DaaG opens that document as text in a separate WWW browser for viewing. Users are not presented with document titles but are allowed to view the DaaG representation, or the full document.

We indexed the document collection by word stems or statistical phrases as documented in our TREC-5 mainline submission and provided a ranked retrieval as described in section 2 above. For the high precision track runs we used no query term or posting list thresholding. The user entered a query into a text box and for these experiments this consisted of whatever search terms the user thought appropriate. The search term weighting was tf*IDF. A WWW interface to our retrieval engine was developed for this track and the code to create a DaaG as the result of running a query, was developed. Encoding a DaaG was not as much of an overhead as might initially seem since our inverted file entries store, for each set of index term occurrences in a document, the location of the first occurrence, the location of the last occurrence, and the number of occurrences, as measured using non-stopword term offsets. These 3 values are compressed to fit into 4 bytes of storage and the file of index term occurrence offsets is used as a "shadow" file on the main inverted file [Kelledy, 1997]. This has the advantage of not doubling the size of the raw inverted file unnecessarily, and this positional information can be used or ignored without effecting retrieval performance.

The code to generate a DaaG was written in Java and uses widgets from the Java Development Kit (JDK) 1.1 in order to present the results. At the time of running these experiments, neither the then current versions of Netscape not Internet Explorer supported this so we used the HotJava browser. When loading the URL into HotJava, an applet is launched which invites the users query into a text box. When the SUBMIT button is clicked, this query text is sent back to a server program running at DCU which calls the search engine to run the query. The resulting ranked list of document identifiers and the positional information of search terms in each of the top ranked documents, is

[1] Naturally, for the version we used in experiments these were real colours, not greyscales.

then sent back to the applet for display. A screendump of a mid-search is shown below in Figure 3.

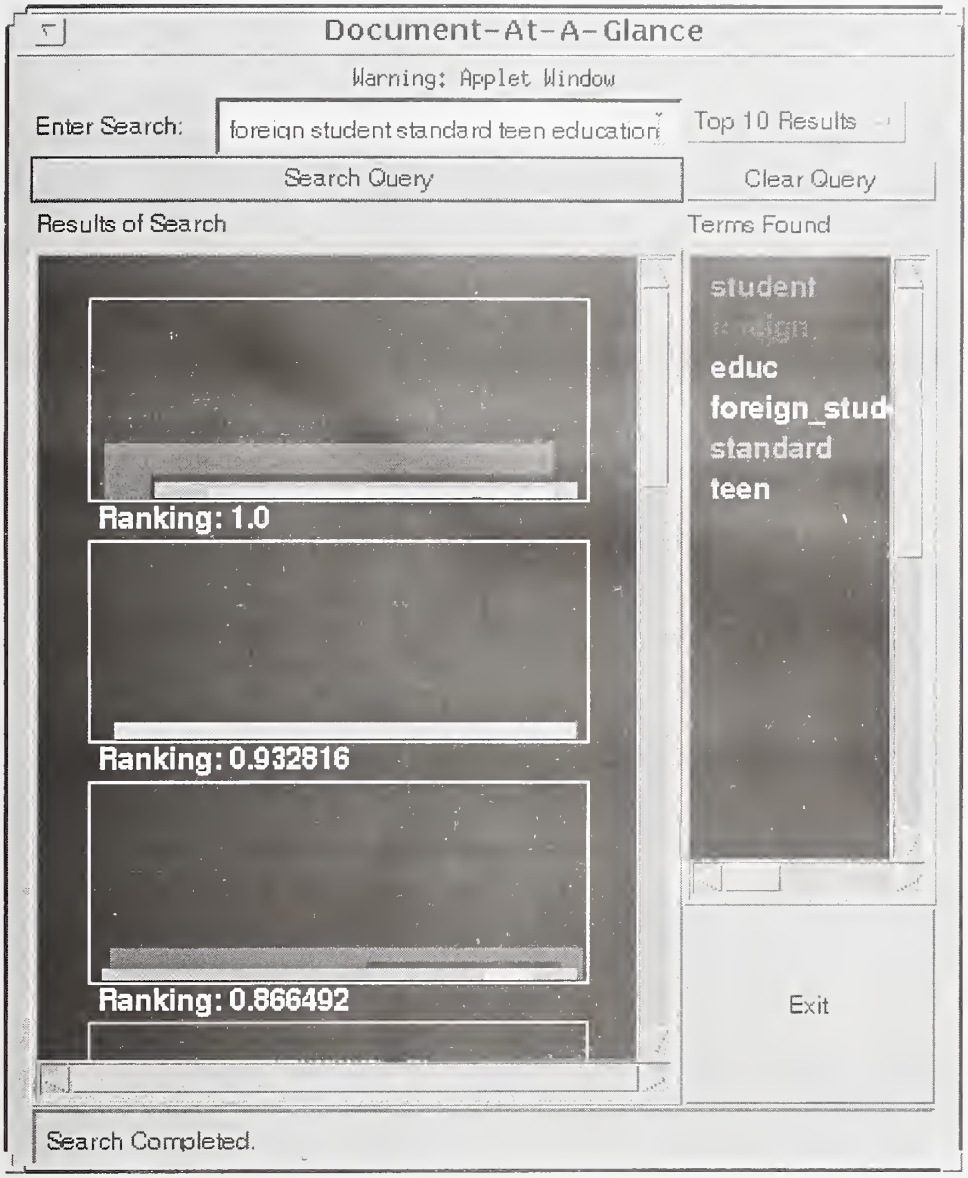


Figure 3: Sample DaaG

Here we see the user's query, which is for topic 346 has generated a ranking and we are looking at the DaaGs for the first 3 documents in the ranking. The user's search terms are "foreign student standard teen education" and the phrase recogniser has picked up "foreign_students" as a phrase. The top-ranked document (or more correctly, a fixed-length window in this top-ranked document) has occurrences of the term student spanning the start of the document through to almost the end and the height of its rectangle (the one at the back) in the DaaG shows it contributes most to the score of this document. For this document the terms contributing to the RSV are, from the back of the Daag to the front, student, foreign, education, foreign_student and there is a tiny occurrence,

probably just once and with a low IDF weight, of the word standard about three quarters through the document. For the document which is third in the ranking, the scoring is again dominated by the contribution of the term student and then education with the terms foreign_student and foreign occurring towards the end of the document. A user would click on this DaaG at some point towards the end of the document and it would be loaded at that position in the document viewer.

We used one subject for our high-precision runs and she ran all of the 50 topics over a period of about 4 days, using only 5 minutes but spacing out the sessions for sanity. The subject was an undergraduate female student (economics major) in her early 20s, with no experience of information retrieval or of using, let alone searching, the WWW, a real naïve user ! She was given a 15 minute overview of the system and its interface during which she ran some dummy searches to become acquainted with the system.

The retrieval results we obtained in our single submitted run in this track were as follows, averaged over all 50 topics.

Avg. Precision @ 10 docs	0.3820
Relative Precision @ 10 docs	0.4031
Unranked-Avg-Precision @ 10 docs	0.0633

We analysed how many relevant documents were found in each of the 50 topics and found the following distribution:

No. Reldocs found	10	9	8	7	6	5	4	3	2	1	0
No. topics	1	3	2	5	4	6	6	4	4	5	10

This table showed that there were 10 topics (of 50) where we retrieved no relevant documents in our top-10 selected. In looking at the output of the search in order to select documents for retrieval, the user examined an average of 30.7 DaaGs. We did not record how many documents were actually viewed but it was somewhat less than this.

In general, these results look poor but like the other 4 participants in this track, we were there to learn. Some of the things we learned were that we need to use more than one subject as she became disenchanted (or brain-dead !) with the task, and we need to use subjects who are more familiar with the technology. While it is a nice idea to use a complete novice, in today's world of information retrieval, web searching has made the user more IR-savvy and the more sophisticated the user, the more we feel our DaaG is appropriate as a tool. Another thing we learned were changes to make to the software to improve it. While it did not crash, because it has to be run from the HotJava browser it is slow and unwieldy to move windows about and to scroll. We created DaaG icons for the top-10 documents at a time and had these concatenated together as a palette which was scrollable, but given the computing environment, slow to refresh. For the next version we will load a DaaG icon onto a fixed place in a window as a document is highlighted. Another feature we will replace is the display of the title of a document, which was missing from the experiments reported here. We were surprised how information-rich the title of a document can be. The final change we will make to the software is to improve the document loading speed, which was loaded as an HTML file in a browser, and was slow.

Our future work with the Document-at-a-Glance will continue once an updated version has been completed. For evaluation however, we will remember that we will be evaluating the DaaG, not the whole system. Thus our future evaluations will compare performance against the top 10 ranked documents and against a retrieval of the best 10 based on previewing documents and their titles only.

5. Triphone Retrieval of Spoken Documents using Fixed Windows

The objective of our participation in this track was to develop the best input parameters to use in a related project we are working on with Ireland's national radio broadcasters - RTÉ. In this related project we are developing a system to search an archive of radio news broadcasts and we use the opportunity of this track to determine parameters related to our search implementation. Our initial searching technique on the RTÉ archive of audio will be on a stream of audio with no story boundaries available. For this reason we decided to do treat the QSDR transcripts as a live stream of text and we removed all story boundaries from the TREC data thus making our task more difficult but closer to reality.

All the story boundaries and tags from the baseline (.srt) and reference (.ltt) transcripts were removed. These raw text files "streams" were then broken into overlapping windows or "documents" of fixed sizes and fixed amounts of overlap. A weight was assigned to each word in the documents based on location within these windows. The identifier of each window was a combination of the file name and percentage of window offset into the transcript eg the document b960610a_23.4. is 23.4% into the file b960610a. The individual windows were indexed by the terms occurring within them. Topics were matched against these fixed-sized windows to find the highest-scored windows using the retrieval engine from our mainline ad hoc retrieval submission which is based on tf*Idf weighting of terms. The document ID of these windows in our ranking were then used to search back into the original baseline or reference transcripts, where a seek of x% was made into the file (percentage in the document identifier). A search was then made backwards to find the first occurrence of a story identifier which was then returned in the ranking as the result.

Our retrieval of RTÉ news broadcasts is based on indexing the audio stream by phoneme units aggregated into triphones[2]. To simulate this in the TREC QSDR track we transcribed the words (from reference and baseline texts) into triphones using a pronunciation dictionary of 160,000 terms in the format of 'computer' represented as *'k ax m p y uu t axr'*. This pronunciation dictionary had been adapted for the RTÉ project by replacing North American pronunciations by Hiberno-English equivalents (colleagues from North America who have participated in previous TRECs will recall the different pronunciations of the term "routing"). Again we broke the "stream" (without story boundaries) into overlapping windows of triphones and indexed these windows by weighted triphones based on their offsets within windows. Topics were also turned into triphones and matched against windows and the stories in which these occurred were returned in the same way as described above.

[2] Our definition of a triphone is of a concatenation of three phones where phones are taken from an alphabet of 41 possible phones

Table 3 shows the combinations of window sizes in words and in triphones, as well as the overlap between these windows, in our submitted runs, on the reference and on the baseline texts.

Text	Run	Representation	Window Size	Overlap	Weighting
Ref	DCU97QSDR-R1	Words	90 words	30 words	Yes
Baseline	DCU97QSDR-B1	Words	90 words	30 words	Yes
Ref	DCU97QSDR-R2	Triphones	120 triphones	40 triphones	Yes
Baseline	DCU97QSDR-B2	Triphones	120 triphones	40 triphones	Yes

Table 3: QSDR Track Runs

Run:	DCUSDR-R1	DCUSDR-B1	DCUSDR-R2	DCUSDR-B2
Mean rank:	9.91	11.80	18.04	14.97
Mean recip:	0.5196	0.5480	0.5022	0.4287
<= 5	34	37	32	28
<= 10	37	39	35	28
<= 20	40	40	39	30
<= 100	46	44	43	35
Not found:	3	4	4	12

Table 4: QSDR Track Results

Overall the runs seemed to produce fairly respectable results. For the reference transcript our average rank on words was about mid-table but on triphones we were almost worst (we were 18.04, max was 18.12). For the baseline recognised text our average rank was 11.8 for word-based and 14.97 for triphone based while the best was 10.11, the median was 17.96 and the maximum was 36.06. It seems therefore that our approach performed better, relatively speaking, on the baseline recognised text with all its inherent noise. Our system failed when there were a lot of short stories close to each other in the transcript as in these cases the wrong story identifier may have been returned instead of the story above or below the correct story or even two stories above. This happened because the actual file space taken up by the tags weren't taken into consideration when the percentages offsets into files were being calculated, i.e. % of how far windows were into a file, and hence when we searched the transcript files we weren't seeking to the exact location of the window and this sometimes caused the wrong story to be returned. This happened in a few cases.

A more significant problem with our approach was that our system didn't do any query expansion on the topics, and there are some topics where the relevant story and the topic texts have no words in common, e.g. topic number 3 Clearly this is something to be considered for future work.

We are encouraged with the results obtained and will proceed to evaluate other window and overlap sizes as well as other weighting functions which consider word/triphone offsets within windows.

6. Overall Conclusions

Dublin City University was delighted to participate in TREC-6 following 4 different lines of action. While somewhat stressful and a bit manic coming up to submission time, we believe that participation in TREC has focused and pushed our work. As a result of this year's participation we are encouraged to continue with the development and evaluation of our document-at-a-glance, the performance figures we obtained in our submissions to the spoken document retrieval track will help us in our own work on retrieval from radio news, and we will return to WST-based retrieval on English texts.

References:

[Hemmje et al, 1994] "LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval", M Hemmje, C Kunkel and A Willett, *in Proceedings of SIGIR'94, Dublin*, July 1994, 249-

[Kelledy, 1997] "Query Space Reduction in Information Retrieval", F. Kelledy, PhD thesis, Dublin City University, February 1997.

[Hearst, 1995] "Tilebars: Visualization of Term Distribution Information in Full Text Information Access", M.A. Hearst, *in Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, CO, May 1995.

[Hearst & Karadi, 1997] "Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy", M A Hearst and C Karadi, in*: Proceedings of ACM SIGIR Conference, Philadelphia*, July 1997, pp.246-

[Smeaton & Kelledy, 1997a] "TREC-5 Experiments at Dublin City University: Query Space Reduction, Spanish & Character Shape Encoding.", A.F. Smeaton & F. Kelledy, *in Proceedings of TREC-5,* NIST (in press), 1997.

[Smeaton & Kelledy, 1997b] "Improving Information Retrieval Efficiency by Search Space Reduction", A.F. Smeaton & F. Kelledy (in preparation).

[Smeaton & Spitz, 1997] "Using Character Shape Coding for Information Retrieval", A.F. Smeaton and A.L. Spitz, *in Proceedings of the 4th International Conference on Document Analysis and Recognition, ICDAR'97, Ulm, Germany*, August 1997, 974-978.

[Spitz, 1995] "Using Character Shape Codes for Word Spotting in Document Images", A L Spitz, *in: Shape, Structure and Pattern Recognition,* D.Dori and A Bruckstein (Eds)., World Scientific, Singapore, 1995

[Spitz, 1997] "Moby Dick meets GEOCR: Lexical Considerations in Word Recognition", A.L. Spitz *in: Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR'97),* Ulm, Germany, August 1997, 221-226.

[Veerasamy & Heikes, 1997] "Effectiveness of a Graphical Display of Retrieval Results", A Veerasamy and R Heikes *in: Proceedings of ACM SIGIR Conference*, Philadelphia, July 1997, pp.236-

Acknowledgement: Part of the work reported here was carried out while GQ was in receipt of funding from FORBAIRT under grant ST/96/707. We also acknowledge support from Greg Grefenstette and the Rank Xerox Research Centre (Meylan, France) for linguistic processing of French topics. Larry Spitz continues to provide advice to us on the subtleties associated with character shape coding.

Appendix I: CLIR TRACK QUESTIONNAIRE:

1. OVERALL APPROACH:
2. MANUAL QUERY FORMULATION:
3. USE OF MANUALLY GENERATED DATA RESOURCES:

None of these questions are applicable to our submission.

4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES:

4.1 Form of the automatically constructed data resources?
[x] Lexicon ... simple wordlist of word form occurrences

4.2 What sort of training data was used to construct them?
[x] Same data as used for searches, subset of data used for searches, about 180 Mbytes.

4.3 Size
[ ] 128,380 entries
[ ] 7 MBytes

4.4 Was there any manual clean-up involved in the construction process?
[x] No clearly benefit could be obtained from cleaning up the wordlist as we have shown for English

4.5 Rough resource estimates for building the data resources (ie. an indicator of the computational complexity of the process).
[ ] __1 hour_____ hours
[ ] __50 Mbytes disk _____ MBytes of memory used
[ ] __50 Mbytes _____ temporary disk space

5. GENERAL

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?
[x] Easily replaceable, _____

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?

[x] Yes, somewhat, a larger lexicon created from a larger sample of French texts

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?

[x] Yes, somewhat,

5.4 Are similar resources available for other languages than those used?

[ ] Yes, we have a similar wordlist for English but much more extensive (300k entries)

Document Retrieval Using The MPS Information Server (A Report on the TREC-6 Experiment)

François Schiettecatte
(francois@fsconsult.com)
FS Consulting, Inc.
1890 Highland Avenue, Rochester, NY, 14618
http://www.fsconsult.com/

1 Introduction

This paper summarizes the results of the experiments conducted by FS Consulting, Inc. as part of the Six Text Retrieval Experiment Conference (TREC-6). We participated in Category C and ran the ad-hoc experiments, producing three sets of official results (fsclt6, fsclt6r and fsclt6t), only one of which was judged (fsclt6). We also produced two sets of unofficial results as part of a database merging experiment that we ran (fsclt6m2 and fsclt6m5).

Our long-term research interest is in building information retrieval systems that help users find information to solve real-world problems. Our TREC-6 participation centered on two goals: to see if automatic query reformulation[1] provides better results than the searcher's initial query formulation; and to continue to evaluate the effectiveness of the document scoring algorithms when searching across multiple databases located on multiple servers.

Our TREC-6 ad-hoc experiments were designed around a model of an end user of information systems who is not a search professional, but one who would occasionally use a system like the MPS Information Server while seeking information in a workplace, or would be familiar with various Internet search engines such as HotBot or AltaVista.

2 Overview of FS Consulting TREC-6 Experiments

In the TREC-6 experiments we set out to answer two questions:

- Does automatic query reformulation provides better results than the searcher's initial query formulation?

We began with the assumption that our information seeker had previous experience using the MPS Information Server and/or various Internet search engines such as HotBot or AltaVista. Although the search interfaces to these systems vary considerably, most systems default to a novice-type search interface[2] that allows a searcher to enter a number of terms and, optionally, apply some sort of operator to relate these terms together such as Boolean operators or phrase searching. To aid the more advanced end-user/searcher, most systems provide a more advanced search interface allowing the user to construct more complex queries as well as some advanced query constructs such as Boolean operators, phrase searching, proximity searching and range

[1] Using relevance feedback.
[2] Usually a single search field.

searching[3]. Typically users don't use these more advanced features, being content to use the novice-type search interface and looking at the first screen of results for a satisfying document[4].

For our first experiment, we did three runs (fsclt6, fsclt6r and fsclt6t), only one of which was judged (fsclt6). We manually constructed queries for all the topics. The searcher was permitted to employ any number of terms along with any search feature offered by the search engine. For this experiment, a single user query was entered for each topic, and a relevance ranked output was generated for each, using standard system features of the MPS Information Server. Once the queries were constructed, they were run producing a first set of results (fsclt6). We then did a second run where the system automatically selected the first two documents from the initial set of results (fsclt6) and applied them as relevance feedback to the search to produce a new set of results (fsclt6r). We did a third run (fsclt6t) using the topic titles to compare the results to the first run (fsclt6).

- What is the effectiveness of the document scoring algorithm when searching across multiple databases?

For our second experiment, the TREC-6 corpus was split and indexed into two separate databases and into five separate databases. We then took the manually constructed queries from the first experiment and ran them across the databases, merging the results into a single ranked list (fsclt6m2 and fsclt6m5). We then compared these results to the baseline run (fsclt6).

The ranking algorithm was initially developed, tested and refined using the corpus and database merging tracks guidelines set out in TREC-4, and were tested more fully in the TREC-5 database merging track.

3 Searcher Model and Guidelines

Because all the runs employed the same query formulations, the same searcher model and guidelines apply to all of them. All query statements for the experiments were constructed by one person. The initial parameters of the searcher 'model' were defined as follows:

- s/he occasionally uses Internet search engines;
- s/he occasionally uses search engines in the work setting;
- s/he may have some search training, but is not a professional searcher;
- s/he dislikes reviewing large search outputs;
- s/he is seeking information to solve a real-life problem;
- s/he may not be a content expert in the topic area of a given question.

3.1 Instructions to Searcher

The following instructions guided query formulation:

---

[3] Such as 'since a certain date' and 'before a certain date', or 'within the last week' or 'within the last month', for example.

[4] It was mentioned in a SIGIR 97 session by Doug Cutting of Excite that users provide an average of 2 terms per search and that only 20% of users request that the second screen of results be displayed after looking at the first screen of results (each screen displays 20 hits). This means that 80% of users either find the document they are looking for in the first screen of results or give up the search.

- prepare a single search statement that will capture the most relevant documents for a given topic;
- use single or multiple terms, employing wild-card capability to capture multiple versions of a word, and/or quotes around several words (e.g., "cardiac arrest") to create a fixed phrase;
- apply Boolean logic as desired, using AND, OR or NOT operators. Create nested statements using parentheses if desired;
- no other databases are available for consultation;
- the total time taken to prepare a single query should not exceed 5 minutes.

3.2 Searcher Training

In preparation for the experiment, the searcher performed training exercises using the TREC-6 training data. First, general capabilities of the system and features of the search engine were described. Then, three topics were selected from the TREC-5 topics by the searcher. For each topic, a query formulation constructed by the searcher was run against the test database in an interactive fashion via a Web based interface. Results were also analyzed using the Trec Eval program. The searcher was allowed to reformulate and re-run training queries as many times as he desired.

4 System Configuration

The MPS Information Server is a commercial full-text retrieval system that runs on a large number of Unix based platforms. Given a user query, the MPS system returns a list of relevance-ranked documents from a database. The system is capable of performing simple or complex term searching, phrase searching and proximity searching using parentheses, wildcards and Boolean operators. Soundex, typographical variation [5]. Fielded searches and numerical range searches are also supported. The system is designed to favor precision over recall when performing searches. Because it supports a number of different protocols, including WAIS-88, Z39.50-V2 (WAIS-V2 profile), STARTS and Gopher as well as two internal protocols, LWPS and Direct[6], the MPS Information Server is capable of responding to search requests from a wide variety of clients applications.

The TREC-6 experiments employed version 4.2 of the MPS Information Server running on a SparcStation 5/110 with 128 megabytes of RAM. Four gigabytes of disk space were set aside and split evenly between data and indices. For the purposes of the experiments, we used a driver application which was built for TREC-4 and was used in TREC-5. Running on the SparcStation, the driver application communicated with the MPS Information Server using the LWPS protocol. This driver application was designed to read TREC topic files, build a query by extracting a specific field, or fields, from the individual topic entries[7], run the queries against the MPS Information Server and save the query results in the TREC result format to a specified file. The results files could then be processed by the Trec Eval program to obtain the precision-recall values for that run.

[5] For example, missing letters, 'color' would also pick up 'colour', and juxtaposition of letters, 'animal' would also pick up 'ainmal'.
[6] LWPS is an inter-server communications protocol and Direct is a protocol which allows for rapid integration into front-end development application tools such as Perl or Tk/Tcl.
[7] In fact the queries formulated by the user are embedded into the TREC topic file and are marked up with SGML tags .

A special parser was built to index TREC databases for the TREC-4 & TREC-5 experiments and was reused, with a few modifications, for the TREC-6 experiments. We created an artificial headline for each documentby concatenating the documentID field [8] and the documenttitle [9]. The rest of the documentswere indexed as plain text, with the SGML tags extracted from the text, and the words stemmed using a plural stemmer. No additional information was extracted from the text except for the word positions to allow phrase and proximity searching if desired by the searcher. All keywords in the news articles were suppressed as required by the guidelines.

While the MPS Information Server's indexing application starts up with a default stop-word list (containing 377 words), it can be made to convert a word to a stop word if that word's total occurrencein the database reaches a specific threshold value. The stop word value, which is site- and collection-dependent, would typically be set anywhere in the range of 20,000 to 500,000 occurrences.For the TREC-6 experiments, it was set at 500,000 occurrencesto retain as many words as possible in the database. This resulted in a final stop-word list of 382 words.

Two databases were created, one containing disks 2 and 4 (the ad-hoc training database) and the other containing disks 4 and 5 (the ad-hoc test database). Each database took approximately 8 hours to build. Their index sizes were about 540 megabytes for disks 2 and 4, and 610 megabytes for disks 4 and 5, from an initial data size of 2.0 gigabytes and 2.1 gigabytes respectively. In addition seven other databases were created for the database merging experiments from disks 4 and 5.

5 TREC-6 Results for FS Consulting Experiments

5.1 First Experiment

In the first experiment, the searcher initially created a single written query for each of the 50 ad-hoc topics. The queries were then runproducinga first set of results (fsclt6). We then did a second runwhere the system automatically selected the top two documentsfrom the initial ranked set of results and applied them as relevance feedback to the search (fsclt6r).

The relevance feedback algorithm employed for query expansion used for the second runis the standard one implemented in the MPS Server. It works by scoring all the terms in the selected documents.The top twenty terms were chosen from that list for further use. This run's final result sets were produced by expanding each original query to include these new terms, assigning weights to the old and new terms, and re-ordering documentsbased on new relevance weights. The relevance feedback can be set to either increase recall or improve precision and the former setting was chosen for these experiments.

This automated query expansion feature was designed as a tool that could be used by information seekers who, having retrieved a large set from an initial search, wish to increase the likelihood that all relevant documentsretrieved were listed in the first 30 or 40 titles in the result set.

[8] The <DOCNO> </DOCNO> field.
[9] Where a title was easily identifiable and available from the document, this was the usually the case for news data, but less so for other data.

5.1.1 Searcher Performance

Training exercises influenced the searcher's query formulation behavior in the following ways:

- he preferred to use the wild-card capability selectively to increase recall, rather than entering multiple forms of a word and diluting the precision of the query;
- he added multiple synonyms, believing that it would increase recall in a selective fashion;
- he kept the queries short;

The following examples are typical formulations. The searcher wrote out the formulations, which were embedded into the TREC topic file without further modification.

Topic 332: "united states" AND "tax evasion" AND investigations

Topic 338: aspirin AND (adverse OR risks)

Topic 344: (email e-mail "electronic mail") AND (abus* spam*)

Topic 349: catabolic anabolic metabolic metabolism glycolysis krebs

Most query formulations employed parentheses, wildcards and the AND and OR Boolean operators. As the examples indicate, not all capabilities of the system were employed (e.g., field searching, proximity searching, soundex,typographical variation and "NOT" operators were not used for example). The bounded phrase was the most common special feature used.

5.1.2 Server Performance

The results for fsclt6 produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       all  fsclt6
Total number of documents over all queries
    Retrieved:    12119
    Relevant:      4611
    Rel_ret:       1300
Interpolated Recall - Precision Averages:
    at 0.00       0.7064
    at 0.10       0.4472
    at 0.20       0.3356
    at 0.30       0.2034
    at 0.40       0.1581
    at 0.50       0.1255
    at 0.60       0.0719
    at 0.70       0.0370
    at 0.80       0.0333
    at 0.90       0.0167
    at 1.00       0.0167
Average precision (non-interpolated) over all rel docs
                  0.1691
Precision:
  At    5 docs:   0.4520
  At   10 docs:   0.3900
  At   15 docs:   0.3467
  At   20 docs:   0.3120
  At   30 docs:   0.2680
  At  100 docs:   0.1642
  At  200 docs:   0.1037
  At  500 docs:   0.0500
  At 1000 docs:   0.0260
```

```
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:          0.2173
```

Overall, for all topics, 28% of the relevant documentswere retrieved from the database and only 11% of the documents retrieved were relevant.

The results for fsclt6r produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       all  fsclt6r
Total number of documents over all queries
    Retrieved:    49001
    Relevant:      4611
    Rel_ret:       1659
Interpolated Recall - Precision Averages:
    at 0.00       0.6847
    at 0.10       0.4703
    at 0.20       0.3571
    at 0.30       0.2414
    at 0.40       0.1665
    at 0.50       0.1040
    at 0.60       0.0436
    at 0.70       0.0239
    at 0.80       0.0040
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) over all rel docs
                  0.1660
Precision:
  At    5 docs:   0.4280
  At   10 docs:   0.3800
  At   15 docs:   0.3400
  At   20 docs:   0.3150
  At   30 docs:   0.2800
  At  100 docs:   0.1682
  At  200 docs:   0.1121
  At  500 docs:   0.0570
  At 1000 docs:   0.0332
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:          0.2142
```

Overall, for all topics, 36% of the relevant documentswere retrieved from the database and only 3.4% of the documentsretrieved were relevant. This last figure is much lower than the same figure for the previous run(fsclt6) because there were many more documentsretrieved in this run(49,001 documents)compared to the previous run(12,119 documents). This increase in the numberof documentsis due to the relevance feedback feature being set to increase recall rather than precision.

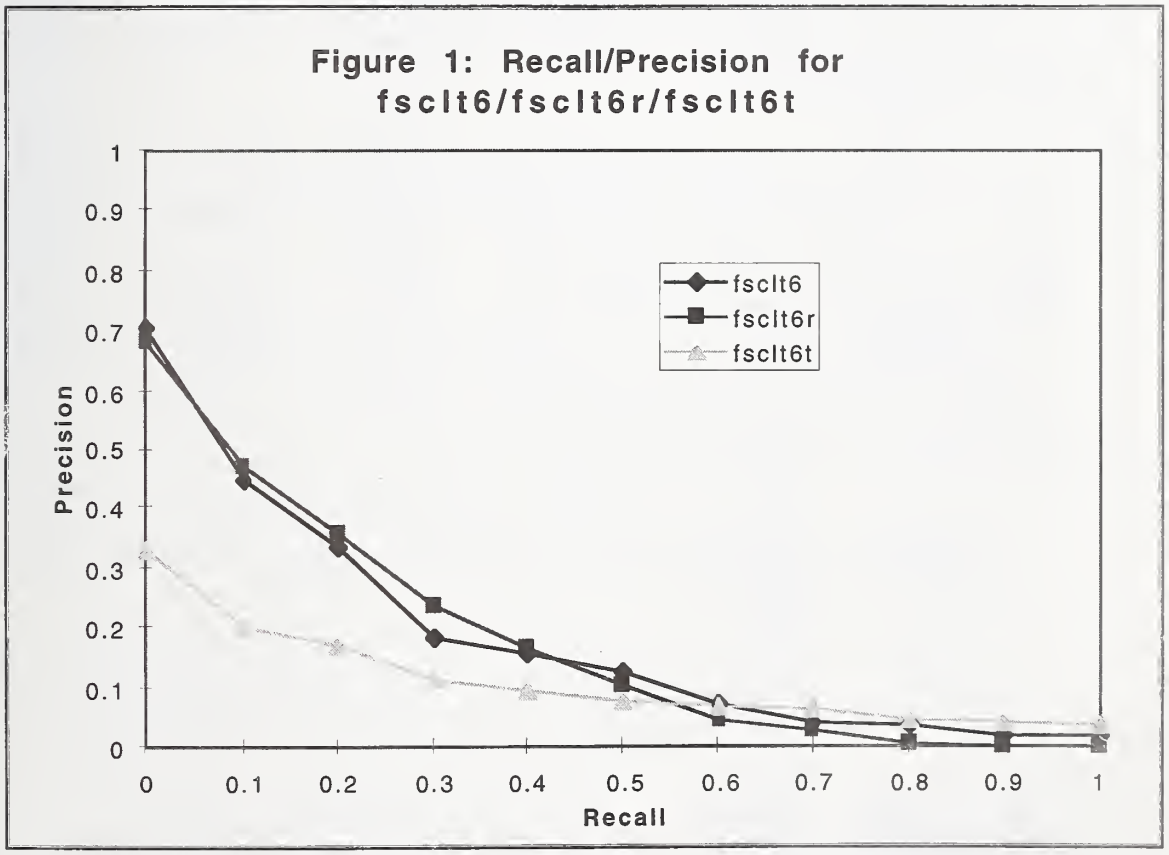
The results for fsclt6t produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       all  fsclt6t
Total number of documents over all queries
    Retrieved:    43156
    Relevant:      4611
    Rel_ret:       1172
Interpolated Recall - Precision Averages:
    at 0.00       0.3340
    at 0.10       0.1992
    at 0.20       0.1718
    at 0.30       0.1125
    at 0.40       0.0943
```

```
    at 0.50       0.0756
    at 0.60       0.0670
    at 0.70       0.0620
    at 0.80       0.0410
    at 0.90       0.0367
    at 1.00       0.0334
Average precision (non-interpolated) over all rel docs
                  0.0958
Precision:
  At    5 docs:   0.1720
  At   10 docs:   0.1680
  At   15 docs:   0.1533
  At   20 docs:   0.1430
  At   30 docs:   0.1233
  At  100 docs:   0.0752
  At  200 docs:   0.0531
  At  500 docs:   0.0336
  At 1000 docs:   0.0234
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.1279
```

Overall, for all topics, 25% of the relevant documents were retrieved from the database and just under3% of the documentsretrieved were relevant. This run is considerably worse than the previous two run strongly suggesting that merely using the topic title as a search string is not a good approach for this search engine.

Figure 1 below show the precision-recall curve for fsclt6, fsclt6r and fsclt6t.



These results would seem to suggest that using automatic relevance feedback would produces better results than not using it, and it appears to improve recall at the expense of some precision.

When comparing these results with our TREC-5 results [2], two differences jump out: the first one is that these runs returned fewer documents overall as well as proportionally fewer relevant documents; the other is that the precision was higher in these results than in the TREC-5 results. This is in line with the design of the search engine which is to favor precision against recall, so when fewer documents are retrieved, precision increases.

5.2 Second Experiment

The second experiment was to measure the effectiveness of the document scoring algorithms when searching across multiple databases. The TREC-6 corpus was split and indexed into 2 separate databases (disks 4 and 5) and into 5 separate databases (cr, fbis, fr94, ft and latimes). We then took the manually constructed queries from the first experiment and ran them across the databases, merging the results into a single ranked list (fsclt6m2 and fsclt6m5). We then compared these results to the baseline run (fsclt6).

It should be noted that these databases were located behind individual MPS Information Servers which were accessed by the MPS Information Server Gateway. This gateway presented the various physical databases as a single logical database to the driver application. The driver application was unaware of the fact that multiple physical databases were being searched and that results sets were being merged, or where these databases were located. In addition, none of the MPS Information Servers were aware of each other's presence, so no collection information was exchanged between them. Each database was searched individually by a single MPS Information Server.

5.2.1 Server Performance

The results for fsclt6m2 produced the following precision/recall figures for all the topics:

```
Queryid (Num):       all fsclt6m2
Total number of documents over all queries
    Retrieved:    13701
    Relevant:      4611
    Rel_ret:       1293
Interpolated Recall - Precision Averages:
    at 0.00       0.7095
    at 0.10       0.4430
    at 0.20       0.3237
    at 0.30       0.2081
    at 0.40       0.1704
    at 0.50       0.1146
    at 0.60       0.0921
    at 0.70       0.0520
    at 0.80       0.0365
    at 0.90       0.0194
    at 1.00       0.0167
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1712
Precision:
  At    5 docs:   0.4440
  At   10 docs:   0.4000
  At   15 docs:   0.3440
```

```
  At   20 docs:   0.3040
  At   30 docs:   0.2607
  At  100 docs:   0.1540
  At  200 docs:   0.1002
  At  500 docs:   0.0478
  At 1000 docs:   0.0259
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2216
```

Overall, for all topics, 28% of the relevant documentswere retrieved from the database and only 9.4% of the documents retrieved were relevant.

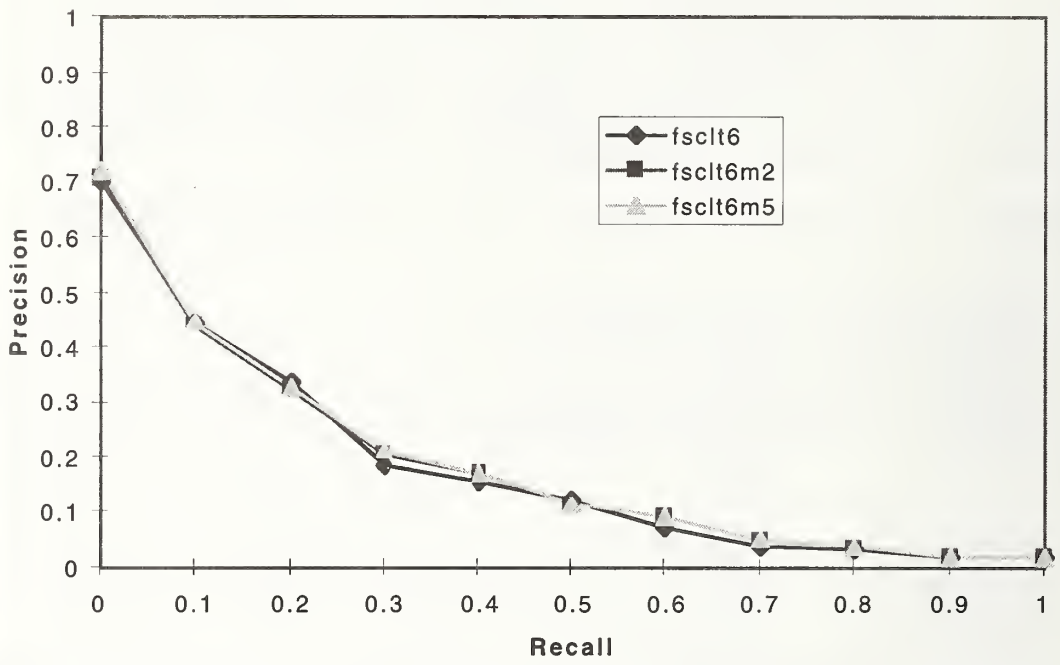
The results for fsclt6m5 produced the following precision/recall figures for all the topics:

```
Queryid (Num):       all fsclt6m5
Total number of documents over all queries
    Retrieved:    13683
    Relevant:      4611
    Rel_ret:       1303
Interpolated Recall - Precision Averages:
    at 0.00       0.7217
    at 0.10       0.4472
    at 0.20       0.3253
    at 0.30       0.2130
    at 0.40       0.1704
    at 0.50       0.1161
    at 0.60       0.0915
    at 0.70       0.0521
    at 0.80       0.0365
    at 0.90       0.0194
    at 1.00       0.0167
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1729
Precision:
  At    5 docs:   0.4520
  At   10 docs:   0.4100
  At   15 docs:   0.3493
  At   20 docs:   0.3110
  At   30 docs:   0.2620
  At  100 docs:   0.1566
  At  200 docs:   0.1015
  At  500 docs:   0.0485
  At 1000 docs:   0.0261
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2225
```

Overall, for all topics, 28% of the relevant documentswere retrieved from the database and only 9.5% of the documents retrieved were relevant.

Figure 2 below show the precision-recall curve for fsclt6, fsclt6m2 and fsclt6m5.

Figure 2: Recall/Precision for fsclt6/fsclt6m2/fsclt6m5



What is interesting to note is that the two merged runs (fsclt6m2 and fsclt6m5) retrieved as many documents as fsclt6, and that the precision is generally the same and even a little higher for certain recall values. This would seem to suggested that the document scoring algorithms works well when searching across databases. These results are essentially the same as the ones we got in the TREC-5 [2] database merging experiments.

The implication of this is that we can segment a very large database across a number of machines to take advantage of parallel processing[10] and be able to present the user with a single, meaningfully ranked, results set. In addition one would also gain in terms of system redundancy where portions of the database would still be available for searching if one of the machines was unavailable due to repairs or maintenance.

6 Discussion of FS Consulting TREC-6 Results

The MPS Information Server is designed to operate in an interactive setting, where quick response and high precision are generally preferable to high recall[11]. Comparing the TREC-6 results with our TREC-5 [2] results really illustrates this. While our TREC-5 results returned more documents overall, the precision was lower. In fact the TREC-6 results mirror the TREC-4 [1] results more closely.

[10] The MPS Information Server Gateway searches multiple databases in parallel.
[11] High recall can be achieved by using relevance feedback; this is the recommended search strategy when high recall searches are required.

6.1 System improvements

While the relevance feedback algorithm works adequately at this point, it is hard not to want better performance from it. In that light we will be runninga numberof experiments this year prior to TREC-7 to fine-tune the relevance feedback algorithms further.

7 Future Work

TREC-6 experiments provided baseline results and in a non-interactive environment and allowed exploration of possible directions for future work. Several themes emerged that will guide our research efforts in preparation for participation in TREC-7, as follows:

- The system will be tuned and improved. The query expansion tool will continue to be tested and revised. Additional relevance feedback algorithms will also be tested.
- Currently multiple database searching is performed using the same search engine and the same ranking algorithms to search all the databases. While this works well in situations where one can use the same search engine to access databases across a numberof systems, it would not work well in situations where multiple different search engines are used[12]. For TREC-7 we plan to develop a merging gateway which would allow us to combine the search results from multiple search engines in a meaningful manner. For this work we plan to use the STARTS [3] protocol recently developed at Stanford as part of the Digital Library Project there.
- Finally we plan to participate in the VLDB track. We had planned to do so for this experiment, but delays and technical problems prevented us from doing so.

8 References

[1] Schiettecatte, François and Florance, Valerie, "Document Retrieval Using the MPS Information Server". In Harman D. (Ed) The Fourth Text Retrieval Conference (TREC-4). National Institute of Standards and Technology Special Publication 500-236, Gaithersburg, Md. 20899.

[2] Schiettecatte, François, "Document Retrieval Using the MPS Information Server". In Harman D. (Ed) The Fifth Text Retrieval Conference (TREC-5). National Institute of Standards and Technology Special Publication,
Gaithersburg, Md. 20899.

[3] STARTS, Stanford Protocol Proposal for Internet Search and Retrieval, Luis Gravano, Kevin Chang, Hector Garcia-Molina, Carl Lagoze, and Andreas Paepcke. Digital Library Project, Stanford University, CA.

[12] As is frequently the case in most 'real world' work environments.

Expanding Relevance Feedback in the Relational Model

Carol Lundquist
George Mason University
Fairfax, Virginia
clundqui@osf1.gmu.edu

David O. Holmes
NCR Corporation
Rockville, Maryland
David.Holmes@WashingtonDC.NCR.COM

David A. Grossman
Office for Research & Dev.
Washington, DC
dgrossm1@osf1.gmu.edu

Ophir Frieder*
Florida Institute of Technology
Melbourne, Florida
ophir@ee.fit.edu

M. Catherine McCabe
George Mason University
Fairfax, Virginia
cmccabe@gmu.edu

Abstract:

In TREC-6, we participated in both the automatic and manual tracks for category A. For the automatic runs, we used the short versions of the queries and enhanced our existing prototype by expanding the relevance feedback methodology to include additional term weighting methods (i.e., the typical "*ltc-lnc*" or "*nidf*" weights) as well as feedback term scaling. We also experimented with eliminating infrequently occurring terms to determine if the relevance ranking scores between documents and queries could be improved by eliminating certain highly weighted terms. For our manual runs, we used pre-defined concept lists with terms from the concept lists combined in different ways. We continued to use the AT&T DBC-1012 Model 4 parallel database machine as the platform for our information retrieval system which continues to be implemented in the relational database model using unchanged SQL.

* This work was supported in part by matching funds from the National Science Foundation under the National Young Investigator Program under contract number IRI-9357785. Ophir Frieder is currently on leave from the Department of Computer Science at George Mason University.

1. Introduction

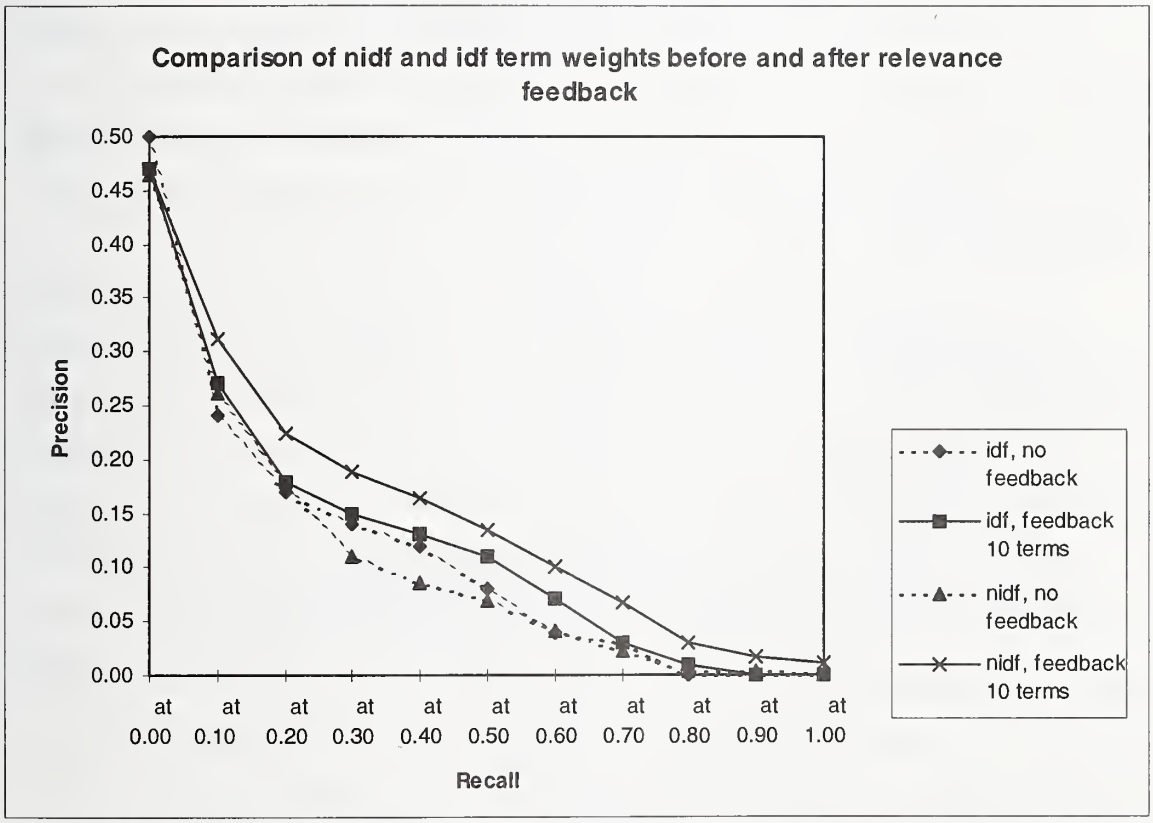
Our work for TREC-6 is a continuation of the work started in TREC-4 when we implemented an information retrieval system as an application of a relational database (RDBMS). We used unchanged SQL to implement vector-space query relevance ranking (Grossman95, Grossman96). The TREC-4 work was expanded upon for TREC-5 when we implemented a basic form of relevance feedback, also using unchanged SQL. For TREC-6, we expanded our relevance feedback methodology to include the lnc-ltc term weights (Singhal96) as well as feedback term scaling. In addition to expanding and improving our relevance feedback methodology, we also experimented with methods to improve the precision and recall scores of our pre-relevance feedback baseline run. To explore the assumption that certain infrequently occurring terms with high collection weights may actually be artificially inflating the query-to-document relevance ranking scores, we experimented with eliminating infrequently occurring terms from the collection. This approach shows promise for improving the baseline scores and has other advantages such as reducing the processing time per query and disk storage space for the document collection.

Our manual runs also represent a continuation of the work started in TREC-4. In TREC-4, we assigned the query terms in up to three concept lists and used general world knowledge to expand the query to include other similar terms not found in the topic. In TREC-5, we continued to use the concept lists and experimented with the use of manually assigned weights to the query terms as well as using manual relevance feedback to identify additional terms. For TREC-6, we augmented our prior work with inexact term matching and an automatically generated thesaurus based on term-to-term co-occurrence. Our first run uses up to three concept lists. To assess the value of using concept lists, our second run uses the same terms and scoring algorithm as the first run, but all of the query terms are placed into a single list. Essentially, multi-concept topics were changed from an intersection to a union of documents. We also introduce a Soundex variation (Celko95) as a tool for expanding the concept lists with similar terms. Finally, an association rule is used to identify co-occurring terms. Full details of these methods and the methods used for the automatic runs are described in sections 3 and 4.

2. Implementation of an Information Retrieval system using the Relational Model

This section provides a brief overview how our information retrieval (IR) system is implemented using the relational model. Full details of the implementation can be found in (Grossman97 and Lundquist97a).

To test the effectiveness of the lnu-ltc or "nidf" term weights over the inverse document frequency or "idf" term weights, we ran several calibration runs on the TREC-5 data to compare the differences in precision and recall both before and after relevance feedback. Figure 1 shows the difference in precision and recall for the two term weighting methods.



--- Figure 1 ---

Using 10 feedback terms, with feedback terms selected by the n * term weight method when relevance feedback was done, and using a subset of documents from Tipster disks 2 and 4 along with the TREC-5 queries, the following results were obtained:

Type of Relevance Feedback	Average Precision	Percent change	Exact Precision	Percent change
idf, no feedback	.0966	----	.1410	----
idf, feedback 10 terms	.1100	+14%	.1421	+1%
nidf, no feedback	.0914	----	.1306	----
nidf, feedback 10 terms	.1400	+53%	.1755	+34%

Table 1 -- Comparison of average and exact precision

An additional benefit to using the relational model for IR is the ability to exploit parallel processing via the DBMS. We implemented an IR system using Teradata's RDBMS on a 4 processor DBC/1012 parallel processing machine. The Teradata DBC/1012 Database Computer is a special purpose machine designed to run a relational database management system using standard SQL.

3. Automatic Results

3.1 First Automatic Run

Our first automatic run used standard relevance feedback similar to that originally proposed by Rocchio in (Rocchio71). For this run, we used the formulas described in (Ballerini96 and Buckley95) to perform an initial relevance ranking to identify the 20 top-ranked documents for each query. We selected the 10 top-ranked feedback terms contained in these documents using the N * *nidf* sort order where N is the number of documents out of the 20 top-ranked documents containing the term and *nidf* is the weight of the term in the document collection. The 10 feedback terms were then adjusted by a scaling factor of 0.5 and added to the original query. The query-to-document relevance ranking was then recomputed using the modified query, and the 1000 top-ranked documents were identified. Further details on the experiments done to determine the optimal number of top-ranked documents and relevance feedback terms to use along with the sort order and scaling for the feedback terms can be found in (Lundquist97b).

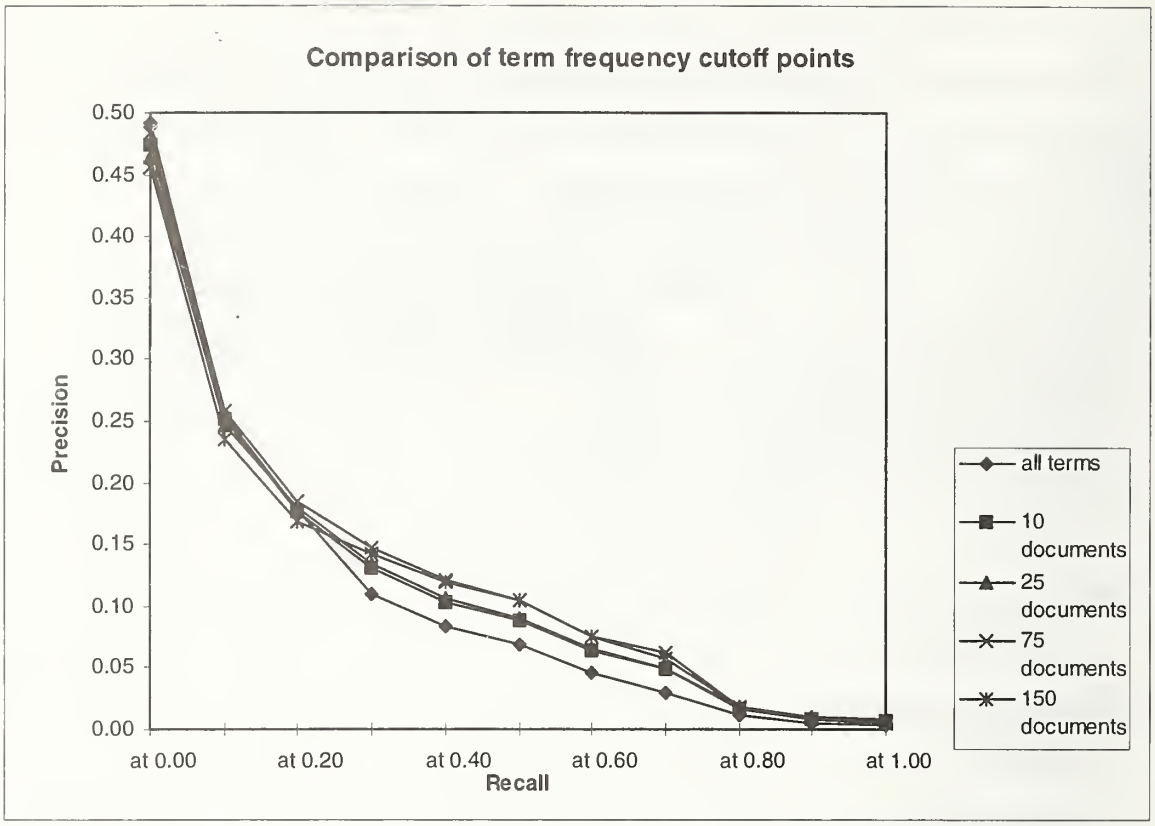
Table 2 shows the comparison of the results from our first automatic run with the other short topic automatic runs submitted and lists the number of queries where we achieved results that were either best, above the median, at the median, or below the median.

	Best	Above Median	At Median	Below Median
Average precision (non-interpolated)	1	29	1	19
Number of relevant documents retrieved	10	23	8	9

Table 2 -- Results comparison for gmu97au1

3.2 Second Automatic Run

In our second automatic run, we did not use relevance feedback. Instead, we attempted to improve the precision and recall scores of our baseline run by experimenting with term frequency cutoff points. To do this, we essentially expanded the stopword list to exclude terms which occurred infrequently in the document collection. To explore the possibility that the large term weights of the infrequently occurring terms may be artificially inflating the relevance ranking scores of documents, we eliminated all terms that occurred in less than 75 documents in the document collection and performed the routine query-to-document relevance ranking. A comparison of the precision and recall levels at different frequency cutoff points can be seen in Figure 2.



--- Figure 2 ---

Using only relevance ranking, *nidf* term weight method, and documents from Tipster disks 2 and 4 with the TREC-5 queries, we obtained the following results:

Terms eliminated if occurring in less than N documents	Average Precision	Percent change	Exact Precision	Percent change
all terms	.0928	----	.1346	----
10 documents	.1032	+11%	.1426	+6%
25 documents	.1051	+13%	.1423	+6%
75 documents	.1149	+24%	.1514	+12%
150 documents	.1083	+17%	.1444	+7%

Table 3 -- Comparison of average and exact precision

Since Tipster disks 4 and 5 combined contain approximately 525,000 documents, 75 documents represents approximately .014% of the document collection. Since infrequently occurring terms make up a large percentage of the number of distinct terms, eliminating terms occurring in less than 75 documents allowed us to reduce the amount of storage required by 26%. Table 2 shows the average and exact precision scores obtained during our calibration runs using

the TREC-5 queries. Based on these calibration runs, eliminating terms occurring in less than 75 documents generated the most improvement (i.e., 24%) over the baseline scores.

The calibration runs on the TREC-5 queries showed that while using term frequency cutoff points did not perform as well as relevance feedback, it did produce a significant improvement over the baseline scores. At the same time, the term frequency cutoff points allowed for a significant reduction in processor time because the second relevance ranking run necessary for relevance feedback was not done. Using term frequency cutoff points also allows overall disk storage to be considerably reduced by eliminating certain terms.

Table 4 shows the comparison of the results from our first automatic run with the other short topic automatic runs submitted and lists the number of queries where we achieved results that were either best, above the median, at the median, or below the median.

	Best	**Above Median**	**At Median**	**Below Median**
Average precision (non-interpolated)	3	17	3	20
Number of relevant documents retrieved	6	17	9	18

Table 4 -- Results comparison for gmu97au02

4. Manual Results

4.1 First Manual Run

Query creation for our first manual run included multiple processing steps. To initially create the manual runs, we examined each topic and selected terms and two word phrases that appeared relevant. We used one pass of relevance feedback and a term-term association list (based on term-term co-occurrence) to give the user potential terms to use in a query. Our user then selected terms and phrases thought to be relevant. The terms were grouped into concept lists based on the assumption that every topic relates to one or more concepts. To be ranked for a given topic, a document had to contain at least one term from each concept list. The remaining terms in the concept list simple increase the similarity measure – they are not all required to be

present in a document. A catch-all list, not part of a concept and not used to qualify documents, had words used for weighting qualified documents. Qualified documents were scored by considering the number of query terms (Q1) shared by a document (X1). The number of distinct terms (K1) tempered results for large documents.

relevance score = $(Q1 \cap X1)/K1$

A Soundex variation was used to expand queries with similar terms. Phrases were assigned two soundex codes, one for each word. Terms and phrases with matching soundex codes were ranked using a similarity coefficient (Pfeifer96) which uses the digram sets for the condition (D1) and result (D2) terms. Digram sets include one leading and one trailing blank to weight the beginning and ending of terms. For example, the word "dog" has the digrams: "_d", "do", "og", and "g_".

similarity coefficient = $(D1 \cap D2)/(D1 \cup D2)$

For a limited number of queries we collected associated terms using an improvement formula (Berry97) used for market basket analysis. Our minimum support was ten documents and the maximum support was 1,000. This deviates from the minimum support of 75 used in the automatic runs.

improvement = p(condition and result)/ (p(condition) p(result))

Finally, we implemented a casual relevance feedback technique. The initial query was run and a list of terms from a few of the top-ranked documents were inspected. If some terms appeared relevant, then they were added to the query and it was run again to produce final results. In most cases, associated and feedback terms were limited to proper nouns. In a few cases, such as topic 349, terms were removed as a result of feedback. For topic 349, the terms "anabolic" produces a large number of documents related to the use of steroids by athletes which did not appear relevant.

4.2 Second Manual Run

Our first manual run, like TREC-4 and TREC-5, used concept lists which create a qualified list of documents that are an intersection of every concept related to a topic. The goal was to create a concise and precise answer to a search request. To measure our assumption, the second run uses the same terms and scoring algorithms as the first run, but instead creates a union of the documents. As discussed in section 4.4, the intersection approach results in better precision.

4.3 TREC-6 Failure Analysis of Manual Queries

Our manual results did not contribute to the judged relevant document collection and therefore our precision and recall scores may be artificially low. Table 5 presents, at various document retrieval levels, the number of documents judged relevant or non-relevant and not judged at all. An interesting measure that may compensate for the lack of relevance assessments is to omit non-judged documents from the measure of precision – this assumes non-judged documents were neither relevant nor retrieved. Precision is then defined as the ratio of the number of judged relevant documents to the number of judged documents at various retrieval levels. Using this measure, the difference in precision is dramatic. Nearly 40% of our results at 100 documents retrieved were not evaluated. By eliminating non-judged results, our precision increased from 19.38% to 34%.

Documents Retrieved	Judged Relevant	Judged Not Relevant	Not Judged	Pct Not Judged	TREC-6 Precision	Precision on Judged Only
at 1	16	19	15	0.3000	0.3200	0.4571
at 5	81	101	68	0.2720	0.3240	0.4451
at 10	150	204	145	0.2906	0.3000	0.4237
at 15	232	293	219	0.2944	0.3093	0.4419
at 20	293	390	306	0.3094	0.2930	0.4290
at 30	408	589	482	0.3259	0.2720	0.4092
at 100	969	1881	1873	0.3966	0.1938	0.3400
at 200	1346	3340	4115	0.4676	0.1346	0.2872
at 1000	2228	7557	17601	0.6427	0.0446	0.2277

Table 5 -- Document Retrieval Level Performance

Table 6 below indicates the query-by-query examination of our first manual run. Interestingly, when over half of the documents were judged, twenty-eight of thirty-four queries were at or above the median. When under half of the documents were judged, only six of the sixteen remaining queries were at or above the median.

Topic	# of Topic Terms	# of Concepts	Judged Relevant 100 docs	Judged Not Relevant 100 docs	Not Judged 100 docs	Estimate Relevant 100 docs	TREC-6 Best 100 documents	TREC-6 Median 100 Documents
301	45	1	4	10	86	33	87	61
302	25	1	50	47	3	51	58	31
303	44	1	10	90	0	10	10	9
304	106	2	41	26	33	52	78	27
305	57	1	2	72	26	11	13	2
306	89	2	7	30	63	28	84	43
307	39	1	20	36	44	35	84	28
308	7	1	2	7	0	2	4	3
309	28	1	0	59	41	14	2	0
310	23	2	3	18	13	7	10	4
311	24	1	90	7	3	91	97	71
312	33	1	9	17	74	34	11	8
313	17	1	74	14	12	78	82	56
314	11	1	16	36	48	32	33	16
315	92	1	6	30	64	28	38	6
316	12	1	34	14	13	38	34	22
317	27	1	5	26	69	28	13	8
318	35	2	3	19	78	30	14	3
319	21	1	13	9	78	40	43	28
320	15	2	5	54	25	14	6	4
321	41	1	4	4	92	35	68	29
322	34	2	16	43	41	30	26	5
323	15	1	34	46	0	34	36	25
324	25	3	81	13	6	83	88	62
325	22	1	7	78	15	12	14	8
326	30	1	24	65	11	28	46	25
327	32	1	3	63	34	15	12	5
328	5	1	9	38	8	12	9	6
329	24	2	20	35	45	35	35	13
330	27	2	18	45	37	31	37	13
331	23	1	17	19	64	39	72	44
332	37	2	56	31	13	60	99	34
333	19	1	26	50	24	34	44	26
334	41	1	13	67	20	20	17	10

335	30	1	45	46	9	48	59	24
336	45	1	5	88	7	7	6	2
337	26	1	33	49	18	39	52	33
338	16	1	4	94	2	5	5	3
339	13	1	7	71	22	14	10	7
340	27	1	29	47	24	37	52	19
341	54	2	10	20	70	34	44	30
342	36	2	9	37	54	27	15	8
343	56	1	14	6	80	41	84	14
344	14	1	4	54	42	18	4	3
345	36	1	7	31	62	28	24	9
346	66	2	1	16	83	29	34	5
347	31	1	27	9	64	49	68	26
348	11	1	2	6	92	33	5	5
349	30	1	23	56	21	30	36	19
350	32	1	27	33	40	41	54	27
Total			969	1881	1873	1606		

Table 6 -- Individual Topic Performance

4.4 Comparative Results

Our measured results varied greatly by topic. Sometimes the results varied because of the complexity of the topic and other times because of the number of documents evaluated. Figure 3 aggregates our results into five groups based on the number of documents in the result set judged for TREC-6. Table 7 shows how far, in terms of the cumulative number of documents, our results were from the median as well as count the number of queries within the group that were at, above or below the median. For the ten most judged topics, nine out of ten had more than the median number of relevant documents retrieved. Similarly, for the ten least frequently judged documents, eight were below the median.

A possible explanation for having so many results unique to our queries is the use of association rules and soundex searches to expand or replace query terms. For example, we did not use a single word or phrase directly from topic 301. Instead we used some of the original terms as input to an association rule to identify the names of individuals, organizations, or activities associated with crime. Table 8 shows all of our query terms and phrases for topic 301. By probably not sharing many topic critical words with other teams, our results for query 301 were largely unevaluated. Table 9 identifies similar terms found by doing a soundex search. We hypothesize that other teams found many of the same results as our team for topic 302 because

we shared topic critical words such as "polio", but ours ranked fairly well because we stacked the query with several similar words which helped weight relevant documents.

The first and second manual runs used the exact same scoring metric and query terms. The initial run used concept lists to intersect documents by requiring the existence of at least one term from each concept list. The second run required only a single term from the entire query to retrieve a document. Any queries having more than one concept list, or a single concept list and additional weighting terms produced different results. Intersections provided much greater precision. Table 10 compares results at various retrieval levels.

Groups Ranked by # of Docs Judged	Above Median	Median	Below Median
Top 10	9	1	0
Upper Middle	5	3	2
Middle	5	1	4
Lower Middle	4	4	2
Bottom 10	1	7	2

Table 7 -- Performance versus Median

abbas musawi	john gotti	enrique camarena	plo gunman
abu nidal	khan younis	ernesto samper	rafael abello
ahmed yassin	lockerbie bombers	evaristo porras	rodriguez gacha
aldo moro	lockerbie bombing	giovanni falcone	royal ulster
alvarez machain	luis ochoa	giulio andreotti	saeb erekat
cali	martinez romero	gravano	shining path
car bomb	medellin	hamas	sicilian mafia
cocaine cartel	miguel maza	hezbollah	sinn fein
cosa nostra	muammer gadaffi	ira gunman	suicide bomber
drug baron	nicola mancino	ira gunmen	toto riina
drug cartel	pablo escobar	islamic jihad	drug lords

Table 8 -- Topic 301

paralytic polio	polio myelitis	polio vaccines
polio	polio outbreak	polio virus
polio cases	polio type	poliomyelitis
polio epidemic	polio vaccine	poliovirus

Table 9 -- Topic 302

Retrieval Level	Run 1 Precision	Run 2 Precision
at 5 docs	0.3280	0.0680
at 10 docs	0.3000	0.0580
at 15 docs	0.3080	0.0680
at 20 docs	0.2980	0.0710
at 30 docs	0.2720	0.0640
at 100 docs	0.1938	0.0492
at 200 docs	0.1347	0.0421
at 500 docs	0.0748	0.0321
at 1000 docs	0.0446	0.0231

Table 10 -- Comparing Intersection and Union runs

5. Conclusions and Future Work

For TREC-6, we focused on improving relevance feedback using the relational model. While the changes in our relevance feedback process significantly improved the precision and recall scores of our results, we still need to look into improved methods of choosing the feedback terms to eliminate the "bad" terms which occasionally surface for some of the queries. Another area we have begun to investigate is raising the precision and recall scores of the baseline run prior to relevance feedback. One of the methods we have found to do this involves the use of term frequency cutoff points and additional work needs to be done to further investigate the relationship between the query-to-document scores and the term weights of the infrequently occurring terms.

For our manual runs, we focused on using new methods such as Soundex and an improvement formula based on market basket analysis to identify query expansion terms. Further work needs to be done to better identify the appropriate query expansion terms.

References:

(Ballerini96) Ballerini, J., M. Buchel, D. Knaus, B. Mateev, E. Mittendorf, P. Schauble, P. Sheridan, and M. Wechsler, " SPIDER Retrieval System at TREC-5," Proceedings of the Fifth Text REtrieval Conference (TREC), sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1996.

(Berry97) Berry, M., and G. Linoff, "Data Mining Techniques," Wiley Computer Publishing, 1997.

(Buckley95) Buckley, C. A. Singhal, M. Mitra, and G. Salton, "New Retrieval Approaches Using SMART: TREC-4," sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1995.

(Celko95) Celko, J., "SQL for Smarties: Advanced SQL Programming," Morgan Kaufmann, 1995.

(Grossman95) Grossman, D., D. Holmes, O. Frieder, M. Nguyen, and C. Kingsbury, "Improving Accuracy and Run-Time Performance for TREC-4," Proceedings of the Fourth Text REtrieval Conference (TREC), sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1995.

(Grossman96) Grossman, D., C. Lundquist, J. Reichert, D. Holmes, and O. Frieder, "Using Relevance Feedback within the Relational Model for TREC-5," Proceedings of the Fifth Text REtrieval Conference (TREC), sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1996.

(Grossmam97) Grossman, D., D. Holmes, O. Frieder, and D. Roberts, " Integrating Structured Data and Text: A Relational Approach," *Journal of the American Society of Information Science*, January 1997.

(Lundquist97a) Lundquist, C., D. Grossman, O. Frieder, and D. Holmes, "A Parallel Implementation of Relevance Feedback using the Relational Model," *Proceedings of the World Multiconference on Systemics, Cybernetics, and Informatics*, July 1997.

(Lundquist97b) Lundquist, C., D. Grossman, and O. Frieder, "Improving Relevance Feedback in the Vector-Space Model," to appear in Proceedings of the Sixth ACM International Conference on Information and Knowledge Management, 1997.

(Pfeifer96) Pfeifer, U., T. Poersch, and N. Fuhr, "Retrieval Effectiveness of Proper Name Searches",*Information Processing and Management*, Vol. 32, No. 6, pp. 667-679.

(Rocchio71) Rocchio, Jr., J. J., "Relevance Feedback in Information Retrieval," Gerard Salton, Editor, The SMART Retrieval System, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

(Singhal96) Singhal, A., C. Buckley, and M. Mitra, "Pivoted Document Length Normalization," *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Ed. Hans-Peter Frei, Donna Harman, Peter Schauble and Ross Wilkinson, SIGIR Forum, August 18-22, 1996.

Ad Hoc Retrieval with Harris SENTINEL

Margaret M. Knepper, Gregory J. Cusick, Kevin L. Fox, Ophir Frieder, Robert A. Killam

Harris Corporation, Information Systems Division
P.O.Box 98000, MS W3-7755
Melbourne, FL 32902
(407) 984-6443

mknepper@harris.com
kfox@harris.com

1.0 INTRODUCTION

Harris Information Systems Division (HISD) focuses on information retrieval support for various Government agencies. In our customers' applications, efficient (in terms of processing time) retrieval rates are critical, and highly accurate but relatively few documents retrieved is the norm. Our SENTINEL approach addresses our customers' needs.

This is HISD first participation in the Text REtreival Conference (TREC). Our team participated in the category C (large data set) manual Ad Hoc track of the Sixth Text REtrieval Conference (TREC-6). Throughout TREC-6, we made modifications to enhance the performance of our system. We improved both the processing time and document retrieval. This paper is an overview our efforts for TREC-6.

2.0 SENTINEL OVERVIEW

2.1 Retrieval Components

SENTINEL is a fusion of multiple retrieval engines, integrating n-gram technology, a Vector Space Model, and a neural network.

2.1.1 N-Gram

SENTINEL employs a n-gram filter based on Julian Youcum's work with least-frequent trigraphs [1]. SENTINEL moves a n-character sliding window over a document while recording the frequency of occurrence of different n-character combinations. A general frequency table is built from a corpus of training documents, representative of the document collection. Relevant documents are rapidly identified by looking for the occurrence of the least-frequent n-gram of a search string in the document. SENTINEL used a 3-character sliding window for TREC.

2.1.2 Vector Space Model

SENTINEL also uses the Vector Space Model (VSM) to represent documents in a n-dimensional vector space. Words appearing in the document training corpus are represented as vectors in the n-dimensional vector space. A vector for each document is constructed based on the terms in a document. A query is considered to be like a document, so a document and query can be compared in the vector space. Documents whose content, as measured by the terms in the document, correspond most closely to the content of query are judged to be the most relevant [2]. The documents are

retrieved through keyword, word clusters (series of words), and example document queries mapped into the n-dimensional vector space. The documents whose vectors are a minimal distance from the query's vector are retrieved.

SENTINEL implemented a new algorithm for our VSM component during TREC-6 which reduced the document representation processing time from three minutes to approximately five seconds per document.

2.1.3 Neural Network

A neural network is used within SENTINEL to train the word vectors in our VSM. The neural network is based on Kohonen's Self-Organizing Map neural network [3], [4]. It is an unsupervised learning algorithm that organizes a high-dimensional vector space based on features in the training data so that items with similar usage are clustered together. This clustering accounts for individual words appearing in close proximity in a document and hence an implied similar or related meaning.

2.2 Ad Hoc Queries

We employed SENTINEL in a multi-level processing approach to manually query for the Ad Hoc topics in the TREC data. A series of queries was created for each of the Ad Hoc topics. In less than two months, our three-person team was able to perform over 900 manual queries over the entire document corpus. Table 2.2-1 shows a summary of the query distribution. The average number of queries per topic was 19. The maximum number of queries for a topic was 90, and the least number of queries for a topic was 6.

For each Ad Hoc topic, keywords and phrases were input to the n-gram component of SENTINEL to obtain rapid information retrieval extraction. This high-level filter yielded initial document screening information. This technique was especially useful for phrases not represented in the training corpus, and hence the vocabulary of SENTINEL's VSM.

Table 2.2-1. Number of Queries

Pass	Number of Queries
n-gram	361
Vector Space Model Pass 1	196
Vector Space Model Pass 2	130
Vector Space Model Pass 3	218
Subset Queries	43
Total	948

Keywords, word clusters, single documents, and document clusters were input to SENTINEL's VSM component as queries. Queries constructed by SENTINEL's VSM can be broadly or narrowly focused, depending on the keywords, phrases, and example documents used in the queries. SENTINEL's VSM was used for high accuracy query retrieval and document scoring. The score is obtained by computing the distance between the vectors representing the query and the document. Document scores ranged from -1 to +1. Negative scores indicated an irrelevant document. Scores for relevant documents[1] ranged from approximately .45 to 1. The closer to 1, the better the document matches the query.

Experiments have shown that SENTINEL's VSM strongest performance results from the use of example documents and document clusters. Initial document examples were obtained

1. Relevant documents are documents we felt met the topic criteria. Irrelevant documents may have been close to the topic, but in our opinion, did not meet the topic's requirements provided in the Description and Narrative accompanying each of the Ad Hoc topics.

from the n-gram filter and topic related articles found on the web. As the passes were completed, top query results were reviewed and identified as relevant or irrelevant. Relevant documents from the query were input to the next pass of SENTINEL's VSM.

New hardware also improved query performance. First, we obtained a SPARC Ultra which increased our RAM from 90 MB to 128 MB. Next, we obtained 16 GB of additional disk space. The new hardware increased the average VSM Pass 1 query processing from 13,000 documents/day to 57,000 documents/day.

2.2.1 Subset Queries

Instead of processing over the entire document corpus for each topic and query, we created subsets for several topics. Subsets were created by the n-gram filter for topics 302 (Polio), 304 (Endangered Species, Mammals), and 347 (Wildlife Extinction) because they contained keyword phrases. Queries were then run over the subsets.

2.3 Ranking

Query results for each topic were processed through multiple methods to enhance the ranking performance. The techniques included relevant document score enhancement and elimination of irrelevant documents and queries. Our algorithm was used to rank query results based on: the number of times the document was selected, highest score, lowest score, and average score.

We reviewed the query results throughout all the passes. As previously mentioned, document scores range from -1 to +1, and the closer to +1 the better the document. As we reviewed queries we set a lower limit for the acceptable documents to reduce the amount of processing required to determine the top 1000 documents. The acceptable lower limit increased as higher rated documents were found. Each topic had it's own acceptable lower limit.

2.3.1 Document Score Enhancement

Relevant documents were used as examples that were incorporated into the next query pass. Originally, the query document wasn't processed as a possible query match. We thought other queries would identify the example documents as good matches and raise the score. However, other queries failed to reintroduce the relevant example documents. The relevant example document was being penalized since it was not acknowledged as relevant and processed. We modified the algorithm to process the example query document as a possible match.

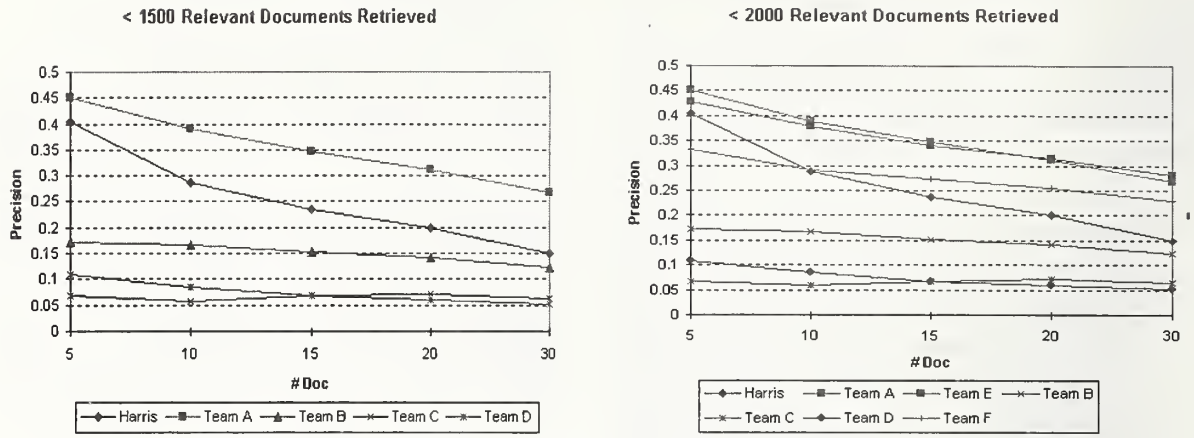
2.3.2 Document Elimination

Top documents were reviewed and irrelevant documents identified. Irrelevant documents were filtered from subsequent queries. Removal of the higher-scoring irrelevant documents allowed lower scoring documents to be accepted on the final result list.

A set of queries was constructed for each topic. For each topic a separate list of query results were maintained. We reviewed individual query results and results from combinations of queries. While reviewing results it became obvious that certain queries did a better job of retrieval then others. Leaving out an irrelevant query permitted more documents from better queries to be added to the final results.

3.0 TREC-6 RESULTS

SENTINEL was designed to yield efficient high precision for a small retrieval set. We compared our precision results with teams that retrieved a similar number of relevant documents. Figure 3.0-1 shows our system maintains a high level of precision for the top 30 documents retrieved. Our precision is higher than other teams that retrieved more relevant



Team	# Relevant Documents Retrieved
Harris	725
Team A	1,300
Team B	1,172
Team C	1,156
Team D	203
Team E	1,659
Team F	1,831

Figure 3.0-1. Harris maintains a high level of precision

documents.

Time constraints and interest-level limit the user to reviewing the top documents before the determining if the results of a query were accurate and satisfactory. The analyst needs the representative documents at the top of the list. SENTINEL retrieves the relevant documents high on the list. SENTINEL permits the user to build and tailor the query as he further defines the topic. The system permits movement from a generic search to a specific topic area through query inputs.

4.0 ANALYSIS

We reviewed several of our ideas to see if they were successful.

4.1 Irrelevant Document Removal

Reviewing the documents for topic relevance is a subjective task. Our analysis revealed that we had marked documents as irrelevant that had been judged for NIST as relevant. It quickly raised the question "Should we remove irrelevant documents?" The answer is "yes", but we will be more judicious about the documents we remove in the future. We re-ran several topics removing and not removing the documents we had previously judged as irrelevant. Removing irrelevant documents improved our precision score and also gave us additional relevant documents on the final list. But we were hurt in one topic in particular for removing too many "irrelevant" documents.

4.2 Pass Improvements

As expected, the results improved with successive passes. Reviews of each of the passes did reveal that several queries retrieved a high percentage of documents in the first pass. This allows the analyst to get good results with the first pass and build upon the query if desired. Table 4.2-1 shows some examples of the percentage of relevant documents retrieved by SENTINEL relative to the final number of relevant documents retrieved for the topic.

Table 4.2-1. Percentage Relevant Documents Retrieved by SENTINEL for Pass 1 and 2

Topic	% Relevant Documents Found 1st Pass	% Relevant Documents Found 2nd Pass	Total # Relevant Documents in TREC	Topic	% Relevant Documents Found 1st Pass	% Relevant Documents Found 2nd Pass	Total # Relevant Documents in TREC
301	73%	88%	474	308	100%	100%	4
303	25%	50%	10	315	31%	89%	67
304	15%	57%	226	324	89%	85%	162

4.3 Irrelevant Query Removal

We are able to select the combination of queries to represent the topic. The data was re-run using all the queries. There wasn't much difference between using all the queries and selected queries.

Individual query examination reveals which queries are retrieving relevant documents. Additionally, a relevant document may be a top document in one query because the scores are lower in the query. Some relevant documents were found reviewing the results from individual queries rather than the results from the entire collection of queries.

4.4 Time Management

We pondered the question - "Should we try and focus on a couple of topics?" or "Should we try to do our best on all the topics?" We decided not to focus on any particular topic and divided the topics among team members. Sometimes we spent too much time on individual topics due to our own interest! Some topics were neglected due to time and difficulty. We needed to accept the fact that not every topic will have hundreds of documents and don't try to find relevant documents that don't exist.

In some cases after we found a large number of relevant documents for a topic, we turned our attention to other topics. This was true for topics 301, 302, 324, and 330. The next question we pondered - "Should we spend more time on topics where we were having success?" We remained focused on our original goal of trying not to focus on specific topics.

In the cases of topics 303, 317, and 344, we couldn't accept the fact that we didn't find a lot of documents. However, in reviewing the dates of the articles in the corpus we began to realize that the data would be limited and quickly turned out attention to other topics.

4.5 Lost Documents

Most of our initial queries for a topic were generic, using keywords as initial query inputs. As we reviewed the documents retrieved by the queries, we interpreted which documents represented relevant topic documents. We proceeded to tailor and modify the query through the relevant documents input to the next SENTINEL pass. Our document selection controlled the query results. The documents started to represent a variety of clusters in our vector set. As we moved into more specific document query examples we found more document clustering, and consequently the document scores started to rise. This caused us to lose some of the generic lower-clustering and lower-scoring documents found in the early passes.

5.0 ANTICIPATED IMPROVEMENTS

5.1 Negative Queries

SENTINEL's VSM component contains a negative query feature. Example irrelevant documents are identified and the system automatically removes this type of document. Modifications to SENTINEL's VSM removal algorithm didn't give us time to fully test the feature.

Negative query removal was performed during the query. However, it needs to be performed during the ranking where the irrelevant documents are identified. We are currently modifying this feature.

5.2 Stronger Integration

Our process was intensely manual. We've already modified the ranking algorithm to perform integration of the results from the n-gram filter and VSM. It allows the user to weight the different retrieval methods based on the individual query, scales results (accounting for higher then average scores from some documents in the n-gram filter), and penalizes documents only found by one method.

5.3 Improved Interface

We are improving the interface to make relevant document classification and query building easier. SENTINEL will be accessed through a web browser.

5.4 Similarity Measure

Currently, SENTINEL's VSM calculates the query score as the distance between the vectors representing the query and the document. New methods for scoring will be explored.

5.5 Multiple Views

This year one person worked on a query for the entire TREC duration. Next year we plan on rotating the queries among team members. We feel this will give us different perspectives and ideas to further develop the queries.

5.6 3-D Viewing

3-D viewing of different document aspects is being explored using Harris VisualEyes tool. This tool enables us to view clustering and document location with respect to the query. This enables a user to locate additional documents relevant to the query, at least by the aspects being viewed.

Figure 5.6-1 shows an example of the 3-D viewer. We tested a query for retrieving stories about the McVeigh trial using CNN web documents. The query keywords were: McVeigh, trial, Oklahoma City, and bomb. Document locations are represented in space by a box, additionally in this view documents determined as relevant by SENTINEL display the document name next to the box. Clustering of documents can be observed in several areas:

- Cluster 1: Clustering of the McVeigh trial stories. Plus additional stories related to the topic, not identified by SENTINEL
- Cluster 2: Bosnia stories dealing with bombing are near the keyword "bomb".
- Cluster 3: O.J. Simpson trial stories appear near the word "trial".

6.0 SUMMARY

We learned a lot in our first TREC. We have a base retrieval system on which we can build and improve. Integration of the n-gram and VSM, an improved scoring algorithm, and TREC scoring experience gives us confidence for an improved TREC experience next year.

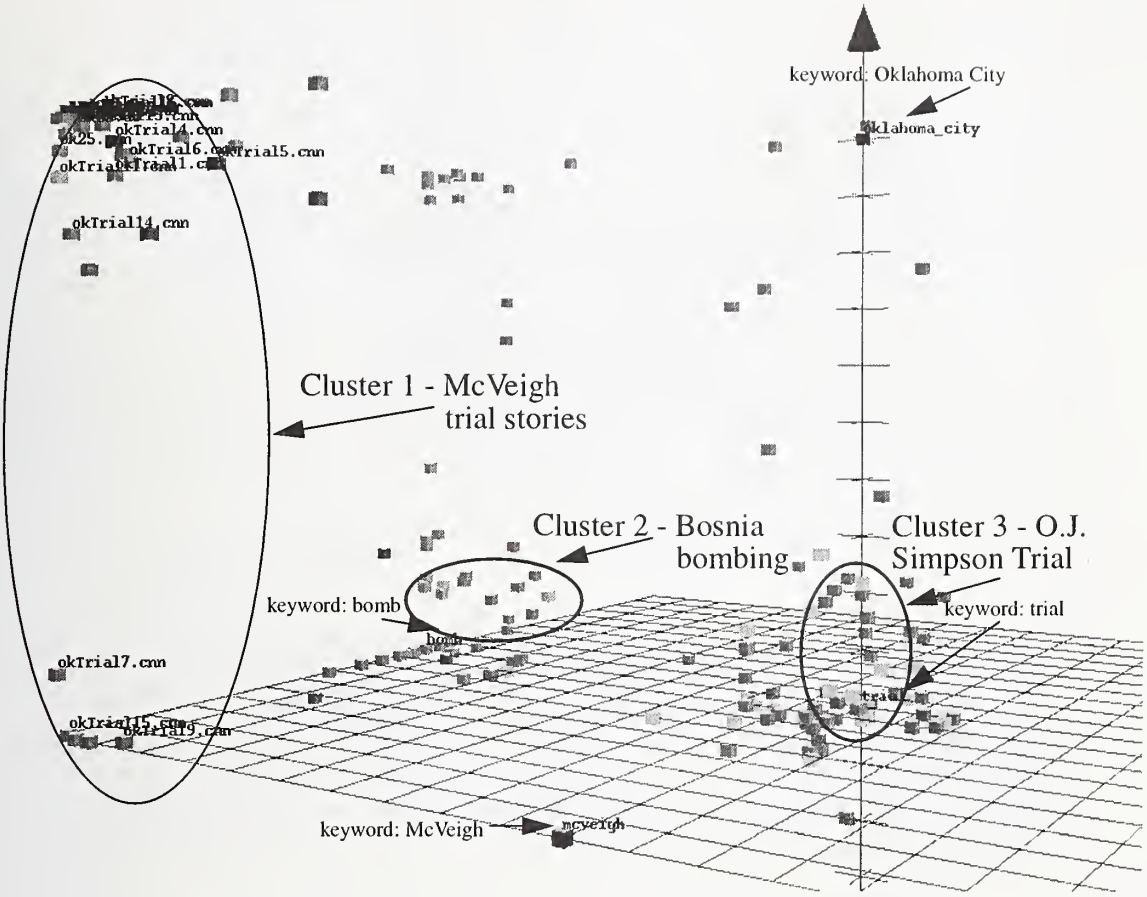


Figure 5.6-1. 3-D viewing of query to find a stories about the McVeigh trial

7.0 REFERENCES

[1] Julian A. Youcum, "High-Speed Text Scanning Algorithm utilizing Least-frequent Trigraphs." IEEE Symposium on New Directions in computing. 1985

[2] David A. Grossman and Ophir Frieder, Information Retrieval: Adhoc Query Processing, Kluwer Academic Publishers, Boston, to appear

[3] Teuvo Kohonen, Self-Organization and Associative Memory. Springer-Verlan, New York, 1984

[4] Simon Haykin, Neural Newtorks: A Comprehensive Foundation, IEEE Press, Macmillan College Publishing CO., New York, 1994

TREC-6 Ad-Hoc Retrieval

Martin Franz, Salim Roukos
<franzm|roukos>@watson.ibm.com
IBM T. J. Watson Research Center,
POB 718
Yorktown Heights, NY 10598

1 Introduction

In TREC-6 ad-hoc experiments we used multi-pass strategy, based on improving the document scores obtained from the Okapi formula [1] by combining them with scores produced by expanded queries, constructed automatically using top ranking documents from the first pass. We have examined various ways of creating expanded sets, as well as computing the scores for words and word pairs contained in them. An application of the same algorithms in the context of TREC-6 Very Large Corpus was also tested.

2 Data Preprocessing

The description fields of the queries and content bearing fields of the documents were filtered and tokenized using a statistical tokenizer. After that the texts of both queries and documents were processed using a morphological analyzer. The analyzer uses part-of-speech tags, obtained from a statistical tagger [2]. Based on the spelling and the word tag, the canonical form of word was found by table lookup. We map 71,472 word forms into 31,434 word stems. The words not contained in the table were kept in their original form. All the words were case-folded after the morphological analysis was done. Hyphenated words were then split into their components.

Before the n-grams were collected for the query sentences, there was a filtering done to remove the common query prefixes, such as "A relevant document would discuss". The filter used a table of such prefixes collected from the previous TREC query sets. All of them and their prefixes were removed if they occurred at the beginning of a query sentence. Such filtering was done for TREC-6 experiments only.

3 Collecting n-grams

Unigram and bigram counts were used for experiments described in this paper. Bigram counts were collected only for the directly neighboring word pairs (phrases), the word order was considered significant. Counts were not collected for the words in a stopword list, containing 514 items.

4 First Pass Scoring

Standard Okapi formula [1] was applied in the first-pass ranking. Each unigram and bigram term in the intersection of the query and document term lists contributed a score of:

$$s = \frac{tf}{c_1 + c_2 \times \frac{dl}{avdl} + tf} \times w^{(1)} \times qtf, \quad (1)$$

where tf and qtf are the document and query counts for a given term, dl is the length of the document, $avdl$ is the average length of the documents in the corpus, $w^{(1)}$ is the inverse document frequency, computed as:

$$w^{(1)} = log(\frac{N - n + 0.5}{n + 0.5}),$$

where N is the total number of documents in the corpus and n is the number of documents containing a given term. In the Eq.(1) we used $c_1 = 0.5, c_2 = 1.5$ for unigram scoring and $c_1 = 0.05, c_2 = 0.05$ for the bigrams. We also decided to set $qtf = 1$ based on experiments with TREC-4. The first pass score was a linear combination of unigram and bigram scores given by Eq.(1), with the unigram scores weight set to 0.8 and bigram scores weight equal to 0.2. The results of these experiments are summarized in Table 1, lines 1 and 2.

5 Query Expansion

We assumed the top 40 documents for each query as ranked by the first pass to be relevant. The documents from these sets were used to establish the new, expanded queries, which were later applied to obtain the second pass scores.

To decide which unigrams should be included in expanded queries, we used a probabilistic model described in [3]. The set of expanded unigrams

contained the words for which the summation of the probabilistic model scores for the top 40 documents was above the 20% of the total score for the highest ranking word.

We also tried a second method for query expansion by using the number of documents containing a given word as an indication whether the word should be included in the expanded query. In this case the word had to occur in at least 20 of the top 40 documents.

Bigrams in the expanded sets were the ones contained in at least 15 of the top 40 documents as ranked by the first pass.

The expanded queries were applied to the document n-grams using Eq.(1) and unigram and bigrams scores were combined the same way as in the first pass. Second pass scores were normalized and combined linearly with normalized first pass scores, using weight set to 0.8 for the first pass and 0.2 for the second pass. The results of these test runs are listed in Table 1, lines 5 and 6.

We also experimented with using the probabilistic model scores directly, both applying the scores of all the words in the document and using only the words from the original and expanded queries, using the probabilistic model scores to decide about the query expansion. The bigram scores were obtained the same way as described above. The results of these experiments may be found in Table 1, lines 3 and 4.

6 Third Pass Scoring

Based on the results of experiments on TREC-4 and TREC-5, we decided to use a three pass strategy for our TREC-6 submission. We trained the probabilistic model using the top 40 documents as ranked by the combination of the first two passes. Expanded queries were created by selecting the unigrams for which the summation of probabilistic model scores for the top 40 documents was above the 20% of the total score for the highest ranking word and bigrams contained in at least 15 of the top 40 documents (i.e. the same way as in the second pass). Third pass ranks were obtained by adding the probabilistic model scores for the n-grams from the original and expanded queries. The final score is a linear combination of the first two passes combined and the third pass, using the weight ratio 90/10. The bottom line of Table 1 summarizes the three pass rescoring results.

			TREC-4		TREC-5		TREC-6	
p1	p2	p3	AveP	P20	AveP	P20	AveP	P20
u,o	–	–	0.2049	0.4030	0.1645	0.2690	0.1697	0.2850
b,o	–	–	0.2275	0.4230	0.1769	0.3050	0.1769	0.3050
b,o	b,ps	–	0.2470	0.4470	0.1846	0.2860	0.1801	0.3030
b,o	b,px,ps	–	0.2513	0.4420	0.1904	0.2910	0.1861	0.2980
b,o	b,cx,o	–	0.2521	0.4500	0.1856	0.2854	0.1788	0.3120
b,o	b,px,o	–	0.2648	0.4660	0.1925	0.2930	0.1819	0.3040
b,o	b,px,o	b,px,ps	0.2695[1]	0.4600	0.1946	0.3050	0.1775[2]	0.2930

u: unigram terms
b: unigram and bigram terms
o: Okapi formula used for scoring
ps: probabilistic model used for scoring
px: probabilistic model used for query expansion
cx: word frequencies used for query expansion
[1] the relevant set used for this third pass run was not obtained exactly the same way as it was done for TREC-5 and TREC-6 experiments, but by using slightly different query expansion scheme, yielding AveP = 0.2652
[2] official TREC-6 result submitted as ibms97a

Table 1: Results of experiments on TREC-4, TREC-5 and TREC-6 ad-hoc.

		baseline		VLC	
pass1	pass2	AveP	P20	AveP	P20
u,o	–	0.0526	0.2670	0.1976	0.3630
b,o	–	0.0564	0.2780	0.2052	0.3620
b,o	b,px,o	0.0626	0.2930	0.2116	0.3860

u: unigram terms
b: unigram and bigram terms
o: Okapi formula used for scoring
px: probabilistic model used for query expansion

Table 2: Results of experiments on TREC-6 VLC.

7 Very Large Corpus

We applied a system similar to the second pass of the ad-hoc run in our VLC experiment, using the probabilistic model for query expansion and Okapi formula to obtain the term scores. The only difference was that the expanded queries were constructed using the documents from CD4 and CD5 (i.e. TREC-6 ad-hoc set) only. This strategy made it possible to run the scoring in a single pass fashion, collecting scores for the original and the expanded queries simultaneously. Unfortunately, the code used for VLC scoring contained an error causing some unigrams and bigrams from the original queries being dropped from the processing. VLC results, summarized in Table 2, contain the average precision and precision at top 20 results obtained using the system after correcting the above mentioned problem, with the original official relevance judgements. The numbers given should be thus viewed as the lower estimate of the real system performance.*

8 Conclusion

We have experimented with various query expansion and scoring algorithms in the context of TREC-4, TREC-5 and TREC-6 tasks. All of the multi-pass strategies improved the average precision as compared to the first pass results. The combination of a probabilistic model used to select the ex-

*We used the query description fields only (as opposed to title and description fields) in our VLC experiments.

panded query words and Okapi rescoring yielded the most significant improvement for TREC-4 and TREC-5, while using probabilistic model based query expansion with probabilistic model scoring was the best in case of TREC-6. Applying a third pass caused a slight average precision improvement in TREC-4 and TREC-5 and relatively large degradation in TREC-6. The overall effect of rescoring diminishes as moving from TREC-4 (our development test set) to TREC-5 and again between TREC-5 and TREC-6.

The Very Large Corpus results were obtained using the first two passes of our ad-hoc system.

9 Acknowledgements

This work is supported by NIST grant no. 70NANB5H1174. We would like to thank Jerome Gros for implementing the statistical tokenizer and Robert T. Ward for contributing to several phases of morphological processing.

References

[1] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford 1995. Okapi at TREC-3. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3).* NIST Special Publication 500-225, 1995.

[2] B. Merialdo 1990 Tagging text with a probabilistic model. In *Proceedings of the IBM Natural Language ITL,* Paris, France, pp. 161-172.

[3] E. P. Chan, S. Garcia, S. Roukos, (to be submitted) A Probabilistic Model for Information Retrieval

IBM Search UI Prototype Evaluation at the Interactive Track of TREC6

Birgit Schmidt-Wesche, Robert Mack, Christian Lenz Cesar
(bwesche @ us.ibm.com, maier @ watson.ibm.com, cesar @ watson.ibm.com)
IBM Thomas J. Watson Research Center, Hawthorne
and
David VanEsselstyn
Teachers College, Columbia University, New York

INTRODUCTION

Our search application used in the TREC6 Interactive Track was developed as part of a User-Centered Design (UCD) program aimed at prototyping UI approaches for using different search technologies being investigated at IBM Research. The search UI we used in the TREC6 Interactive Track focused specifically on four UI issues:

1. The value of web-like +/- query syntax, provided in the sense of the tacitly accepted web standard (with '+' meaning that a term must appear in result documents, and '-' meaning a term must not appear in the returned documents).
2. Support for query refinement using additional terms generated by a Context Thesaurus (CT) program (see Cooper and Byrd (1997)).
3. Display options such as the highlighting of query terms in a result document.
4. Ability to save relevant documents in a separate browser across queries.

The basic components of our search prototype were the IBM NetQuestion search engine (IBM, 1997), the CT component (both running on AIX servers), and a Java application UI (Win95 client) consisting of four tilable windows with the following functions:

1. Query Window with a text field for forming queries and a list of terms related to the user-defined query terms.
2. Result Window, with a list of document results, shown in order, a checkbox for selecting a document, asterisks indicating relative importance ("5-star" rating system), and the document title.
3. Document Window, showing document content, with query terms highlighted.
4. Saved Documents Window, listing documents users selected to save in the Result Window or the Document Window.

When a document is selected for viewing, the title in the Result Window changes from black to blue to indicate that the document has been viewed. Documents can be saved to a separate browser by clicking on a checkbox next to the title in the Result. The "related terms" panel in the Query Window actually showed two levels of related terms: one list displays all terms that co-occur with the query terms in the document collection, and a second window pop-up when the mouse moves over a first-level term. This popup window shows second-level terms related to the selected co-occuring term. The popup window terms change as users move the mouse pointer over co-occuring terms in the first-level list. Both sets of terms are prefixed by check boxes, which, when clicked, add the co-occuring or related terms to the query terms. Unchecking removes the terms from the query specification.

NetQuestion, a search engine especially designed for web search, combines Boolean with Free Text search functions. The '+'/'-' query syntax was designed to exploit these two underlying functions with

each query. What we refer to here as the Context Thesaurus component actually consists of a number of components, each focusing on the various degrees of relatedness between query terms. The type of relationships are determined by frequency, degree of vicinity, and specific syntactic structure in which any two given terms occur to each other in the documents of the collection to be searched. For a more detailed discussion, see Cooper and Byrd (1997).

TEST PARTICIPANTS

We hired five temporary employees from an employment agency used by IBM Research. Appendix 1 summarizes the characteristics of the four participants we used in the evaluation. We rejected one participant because the logging program for our experimental prototype failed, and we lost this person's data. Two participants were "under 40", and two "over 40", and we had one male and one female in each age group. While we had access to professional librarians from another source, we chose to use people from the temporary agency. We specified that these people should be computer literate, and that experience with the World-Wide Web (including Web search) was desirable. We were interested in how usable and useful our prototype UI was for ordinary computer literate people, and not professional librarians. Participants generally conformed to these requirements, although one participant had limited experience with a mouse-based interface, and searching on the Web.

EVALUATION METHOD

We followed the prescribed test procedure with two exceptions noted below. As per the prescribed method, we adhered to the following 10 steps:

1. The participants filled in the questionnaire in which they were asked about their educational background, and their experience with other online search systems and with mouse-based applications.
2. Experimenters read the test instructions to the participants. They could keep them for further reference.
3. Participants worked through the tutorial of the system whichever they had to deal with first (either the experimental system or the control system). The tutorial handout also was available for further reference during the experiment.
4. Experimenter read the TREC6 experiment instructions to the participants.
5. The participants started each task, first reading the topic descriptions and then searching for specific aspects as requested in the narrative and aspects description, writing down on paper the aspect they had in mind when they saved a document.
6. After each 20 min. task, the participants were handed the questionnaire in which they were asked to judge how easy it was to deal with this task.
7. After having completed the first three tasks we took a break.
8. After resuming, the participants worked through the tutorial of the second system, performed the searches, and filled out the questionnaires as before for the first system.
9. Following the three searches of our experimental system, participants completed a questionnaire in which they were asked to evaluate this system.
10. Finally, the last questionnaire asked the participants to compare the two systems in a couple of respects.

In addition, however, we deviated slightly from the overall method in two ways. First, our system hung up during several searches for various reasons. In all but one case, we were able to quickly restart the application, and allowed the subject an additional minute of search time at the end of the session. One

case was more severe. But since the hang-up occurred during the first three minutes of that search session, we stopped it, restored our search system, and resumed again, allowing only the remaining time of 17 minutes to complete that search. We did not reject the data from these participants, although, had we had more time to conduct the evaluation, we might have replaced these participants.

The second deviation is that we sat with participants and allowed them to ask us brief "help" questions which we also answered briefly. If participants looked puzzled or voiced confusion, we also asked them to describe their problem and then we helped them if we could. This kind of interaction is more common in explicit "formative" evaluations where experimenters are more interested in what participants are thinking about as they use a system, perhaps for purposes of developing a help system, or discovering system problems, than in gathering quantative data under strictly controlled conditions for benchmarking purposes (see Landauer, 1989).
We also video-taped the computer screen as participants worked, using a camera in the same room as the participants worked and the experimenters sat.

RESULTS

Here we briefly introduce the quantitative and qualitative results of the evaluation, and briefly introduce specific results requested by the TREC6 Interactive Track methodology, summarized in Appendices 2 through 6. We also contrast in somewhat more detail the search aspects ("topic keywords") our participants identified to those identified by NIST from the collection of submitted documents.

I. Recall and Precision: Quantitative Performance

NIST pooled data from each participating site, and carried out various statistical analyses over the pooled data. There is not enough data from individual participant site evaluations to analyze performance at this level. The experimental design, independent and dependent measures, and statistical analysis model are described in detail at the NIST Web site (general TREC home page at 'http://trec.nist.gov/'; TREC6 Interactive home page at 'http://www-nlpir.gov/~over/t6i/'), and in the overall proceedings of which this report is a part, and we do not repeat this here. Instead we summarize a few key design points and results, and draw conclusions with respect to our prototype performance.

The design is a Latin Square design, which means that the design is an incomplete orthogonal design: Search System is blocked with respect to Search Participant and Search Topic. The statistical analysis is based on derived dependent measures defined by the difference between a Participating Site Search System, and a control system provided by NIST (ZPRISE), for recall and precision. This difference in turn is defined by pairing performance on different Search Topics, that occur in the same order within a test block. Recall is actually *aspectual recall,* and is defined as the proportion of topic aspects identified by searchers, in relation to the total amount available in the corpus, as determined by NIST for a given topic. Precision is defined as the proportion of identified aspects to the relevant topic aspects, as determined by NIST. Again, the details are best understood by consulting the NIST Web site.

Essentially, the design and statistical analysis address the question: For each Participating Site, is there a significant difference between the Search System (for that site), and the NIST ZPRISE control system, with respect to precision and recall? Questions about the effect of Search Topics, and Site Participants of course can be addressed only at the level of pooled data over sites.

The analysis of variance (ANOVA) performed by NIST indicates that there are main effects of Participating Site, Search Participants, and Search Topics for the recall dependent variable. A

comparable analysis has not been carried out for precision. However, while there is an overall effect of Participating Site, most Search System differences by Participating Site are not statistically significant. For example, in our case (the IBM Participating Site), performance was better on the the ZPRISE control system than our prototype, for both recall and precision: Mean recall for the 12 Experimental-Control differences across 3 topic x 4 participant blocks = -0.114, and mean precision for the 12 Experimental-Control differences across 3 topic x 4 participant blocks = -0.019. However, the 95% confidence interval for both mean differences contains zero. And this was the case for all 10 Participating Site comparisons for precision, and 9 of 10 comparisons for recall. Nonetheless, we suspect that with more data points, the difference, for recall in any case, would attain significance. Hence, we conclude that our experimental system does not provide a more effective search UI than the ZPRISE control, and that we have improvements to make. The next section points to some details of query and browsing performance that may guide these improvements.

Two other quick points: First, there was a significant main effect of Search Topic. That is, search topics varied widely in difficulty: the "International Art Crime" topic (322) was hardest, the "New Hydroelectric Projects" topic (307) was easiest across Site Participants. Therefore, while the data are too noisy to statistically distinguish differences between Experimental and Control systems across sites, the differences between Search Topics are large enough to achieve statistical significance. With respect to the search systems and search topics evaluated here, differences attributable to Search Topic difficulty are a much more important factor in relative performance than differences in Search Systems. Finally, there was also a significant main effect of Search Participant. That is, people vary widely in their search performance (measured in terms of recall and precision). This is characteristic of behavioral studies. During the NIST Workshop on the Interactive Track, it was estimated by one site representative that more than four Search Participants would be needed (the minimum prescribed by the experimental design) to achieve enough statistical power for the Search System differences to achieve statistical significance, given the error variance in the data, and the experimental design. Consequently, future evaluations need to use more test participants, and/or use a more powerful experimental design.

II. Query and Browsing Performance

Appendices 5 and 6 summarize frequency of certain query and browsing activities involving our experimental system and the ZPRISE control system. Appendix 5 shows for each Search Participant, the number of queries submitted for each Search Topic, the number of times the Context Thesarus related terms function was used, average number of terms per query, number of documents viewed, aspects recorded, etc., along with aspectual recall outcome for each Search Topic. The tables in Appendix 6 summarize performance over Search Participant, and Search Topic.
The tables indicate that search participants using both the ZPRISE control and experimental systems generated multiple queries, and iteratively refined them by adding or deleting terms. In the case of our experimental system, all participants invoked related terms (generated from the Context Thesarus) at least once, although they tended not to modify query content using these related terms on subsequent search topics. Of course the ZPRISE control system did not provide related terms, or a "+/-" query language, so these activities are only represented for our control system.

The frequencies of activities that are comparable across systems vary across participants, but are actually comparable across our experimental and control systems, with one exception. We had some intense browsers (up to 105 result documents per topic), as well as quick scanners; some used only 2-3 query terms, while others entered up to 10 terms per query. With respect to aspectual recall, two of our searchers did better with the control system, while the other two did better with our experimental system.

Looking at the average recall per topic, we see that all four participants did relatively well on the final topic 339 ('Alzheimer's Drug Treatment'), compared to other topics.

The only striking difference between search systems and search participants can be observed with topic 303 ('Hubble Telescope Achievements'). Two participants using ZPRISE scored significantly better than the other two using our experimental system. It turned out that several "Hubble" documents in the NIST corpus are very short, and our experimental system ranked these more highly than longer documents which provided more aspects for the search topic. Compared to the ZPRISE result list, participants using our experimental system had relatively more difficulty identifying relevant aspects from the result list.

It is not yet clear how differences in these activity profiles account for the difference in recall and precision performance between systems at our site, or across sites. The "+/-" syntax and use of related terms involved extra activities and time not represented in the control system, that would be compensated for only if these activities ultimately resulted in better recall and precision for search topics. At this point, we cannot claim that either feature provided for more effective search in an overall sense, and we do not know whether the *functions per se* are problematic, or the *UI implementation*, or both.

Finally, the data pooled by NIST does not include information at the level of categories we summarize in Appendices 5 and 6, so we cannot compare this level of analysis to other site participants.

III. Summary of Data Requested by NIST

Appendix 1 summarizes the participants' characteristics, in particular their online search experience. Appendix 2 provides the users' judgements of the six topics of the experiment. Appendix 3 provides the questionnaire results for the experimental system, and Appendix 4 the results of the final questionnaire comparing NIST ZPRISE and the experimental system.

Finally, in Appendix 7, we provide the full transcript of participant S1 for the search scenario topic 326 "Ferry Sinking". This participant carried out 9 query iterations, consulting the CT related terms twice, and adding one related term in one query iteration. We note that the participant spends a lot of time expressing questions about the "+/-" syntax for queries. She also repeatedly expresses uncertainty about the type and amount of aspects to save.

IV. Individual Differerences in Aspects Identified by Participants

The aspects our participants identified are largely identical with what the NIST evaluators judged as relevant aspects. But there were some individual differences in how the aspects were recorded. These differences may imply different search strategies, and may also suggest a need for clearer instructions about what constititutes a "successful" search.

1. Brief keyword descriptions versus more detailed descriptions

While two participants quickly described the aspects they identified by simple keywords, the other two chose to describe the aspects in more depth. For example, for the topic 339 'Alzheimer's Drug Treatment', one participant just noted 'Cognex', or 'Tacrine', while another wrote down 'Cognex shows good results for FDA approval', or 'American Medical Association finds good hope in drug Tacrine'. Obviously, the former shorter descriptions were closer to the descriptions in NIST's outline.

2. Which aspects to look for?

The narrative for the topic 326 'Ferry Sinkings' read as follows: 'To be relevant, a document must identify a ferry that has sunk causing the death of 100 or more humans...' Instead of looking for various ferry sinking events, one of our participants looked for only one sinking event, and then concentrated on saving different aspects of this specific sinking event (in this case the Estonia ferry sinking in the Baltic Sea). When asked, the searcher explained that he was just following the task instructions in which it said to'...identify a ferry...'.

3. Is recording aspects a realistic aspect of search?

Whether described in brief or in a detailed way, all four participants felt the task of writing down the aspects they identified as an additional task burden. They had difficulties incorporating this task with the general search task, found it distracting, and had to be reminded of it a number of times.

4. Not enough time to search for all aspects

Only one of our participants was content throughout with the 20 minute time limit for a given search session. The other three believed that they were not finding all the aspects possible, and complained at one time or the other about not having had enough time to do so.

DISCUSSION

First, we give a tentative assessment of the performance of our participants, and which factors may have caused it. Then, we comment on concerns with the TREC6 Interactive Track methodology.

I. How did participants perform using our experimental search UI?

Searching is difficult in general, and judgments about the usability of our specific prototype system must be relative, of course, to other systems. We participated in the TREC6 program to get access to such comparative data.

However, from a qualitative point of view, we note that the search functions we provided, specifically the "+/-" syntax, and the related terms from the Context Thesaurus likely require more experience on the part of searchers to be used effectively. In terms of a system evaluation, this means evaluating longer-term use of these features and/or using more experienced search participants. There are also UI improvements we can make (e.g., performance), but even with these we suspect that there will be a learning curve for all but the most experienced searchers such as professional librarians. Also, adding new functions like related terms to basic search consumes extra time, and so must provide more benefit than doing other things, such as making another query or browsing more documents, or whatever. We had obviously hoped that our participants' performance would improve with these features compared to a system like the control system ZPRISE, which does not provide such features (at least not in the version that was adopted for the experiment). But it is not clear that this was the case. Also, participants may need to develop search strategies for when to do regular keyword search, and when to take the time to use additional features, like using the CT related terms. Once again, this points to the need to evaluate usage over a longer-term, or find radically usable implementations of these new techniques whereby users can immediately make effective use of the features with no training or practice.

Beyond potential UI issues for the features we included in the prototype UI, we note again that our search participants tended to be less experienced than those used by other site participants, and this may have led to some unproductive search strategies. One participant in particular had special trouble making effective queries because of his strategy of creating verbose query specifications, using many non-relevant terms. And, as we also noted earlier, we invited participants to ask questions (treat the experimenter as "help" system) which led to some level of conversational interchange. Lack of experience might account for the overall lower performance on both the control and experimental systems at our site, although the difference in performance between these two systems could still reveal one system as better than another.

We emphasize that the development of our prototype search UI is part of a longer-term design and evaluation process. We evaluated the prototype in order to iteratively improve the UI, and to provide feedback about various problems and options to developers. We note that independent of the quantitative results just summarized, the evaluation of our prototype UI provided much useful "qualitative" feedback, positive and negative about various aspects of the UI and the underlying search technology components. This feedback is of interest to our internal research and development colleagues, and will guide further refinement of the search UI, as will the overall recall and precision benchmarks of the TREC6 program.

II. TREC6 Interactive Track Goals and Methodology

We observe that in addition to the learning curve for the experimental system's UI and search functions, the TREC6 Interactive Track methodology also adds to the learning curve, e.g., reading the instructions, recording aspects, the format of the search scenarios. Streamlining the methodology might also make the evaluation's tasks easier for the participants. This section makes suggestions towards this end.

1. The Goal of TREC6 Interactive Track

As we noted above, we learned a lot about the search technology, and UI decisions we made independent of how our experimental system performed quantatively. The latter quantitative benchmarks established by other TREC6 participants will give us useful targets to achieve in the future. However, is this the only goal of TREC6 Interactive Track? Is there a way that specific search features and functions can be assessed in the context of overall performance? In some cases, of course, these features and functions may be proprietary but over time, it seems to us that the TREC6 community may want to contribute search UI insights to the larger Human-Computer Interaction community. Examples from our experimental prototype include a scratchpad for saving documents and lists of related terms for query refinement.

2. The Search Collection and Search Task

The Web sets users' expectations about what kinds of information can be searched for. The "Financial Times" collection is limited in timeliness and in topic coverage. Are there issues in generalizing what we learn about search from search tasks applied to this collection? For example, search on the Web leads to documents that can be browsed using hyperlinks. Hyperlinking and document browsing add new issues to searching for information.

3. A Critical View on the Written Test Instructions

We believe the written instructions used in the test could be improved and we identify potential areas below. Some items are repetitions of above mentioned ones, but are summarized here for completeness.

Practically **all** types of written instructions, including tutorials and help text (of the experimental system) gave rise to misunderstandings by at least one of the participants, but here we will concentrate on the experiment instructions which contained the general task instructions and the topic instructions, since they are the ones of general interest in this context.

3.1 Experiment instructions

a. How to record aspects
As mentioned above, some of the searchers rendered very detailed aspects descriptions. The instruction 'please write down a word or short phrase to identify the aspect - enough to keep track of which aspects you have found' just left too much room for different interpretations.

b. Misinterpretation of the instructions' example
The general task instruction provided by NIST presented an example to illustrate a typical aspects search that resulted in saving eight aspects (in five documents). As a consequence, two of our participants were ready to stop their search sessions after having found eight aspects. Only after intervention of one of the experimenters, they could be convinced to continue to find more aspects.

3.2 Topic instructions (description, narrative, aspects)

a. Not precise enough
Above we already mentioned that in one case the indefinite article 'a' was misunderstood as the numeral 'one'. Since the narrative required documents that '...identify **a** ferry...', this participant identified one specific ferry sinking event, and then searched for different aspects hereof. Not even the description in the aspects section could prevent him from adopting this view.

b. Too long and detailed?
Some of the topic instructions seemed to be too subtle to clearly draw a line between relevant and irrelevant. Especially the narratives of the topics 307 'New Hydroelectric Projects' and 303 'Hubble Telescope Achievements' seem to have confused some of our participants, who despite the exclusion specifications saved documents on aspects specified as irrelevant. On the other hand, we also observed the case where a participant tried to strictly follow the rules, but taking the topic instructions too literal by only saving documents that complied with **all** of the listed requirements. For example, only when a document was explicitly mentioning the name of a drug, its manufacturer, as well as its success rate, would this participant save this document for the topic 339 'Alzheimer's drug treatment', event though the last item, the success rate, did not seem to be imperative.

4. Controlling characteristics of Evaluation Participants: Should we Agree on Common Searcher Characteristics?

As we noted above, we hired the kind of people we expect to use IBM search technology in Web search services typical on the Web, as compared to professional librarians. However, we did not control or even measure the wide variety of individual differences we know exist between people. There were early, and inconclusive discussions amongst TREC6 Interactive Track participants about whether to provide guidelines on participant selection. We think there should be guidelines, and perhaps some standard measuring tools to address psychological attributes that may influence facility with electronic search systems. An example might be measures of computer literacy, or spatial ability for some UI features.

ACKNOWLEDGEMENTS

We are grateful to Jim Cooper (IBM Research) for letting us adapt Java code components for our prototype, and for advice on various aspects of the server programs used. We are also grateful to Eric Brown and Herb Chong (both of IBM Research) for helpful advice on the use of various server programs.

REFERENCES

Cooper, J. and Byrd, R. (1997). Lexical Navigation: Visually Prompted Query Expansion and Refinement. Proceedings of DIGLIB97, Philadelphia, PA, July, 1997 (pp. 237-246).

IBM (1997). Information about NetQuestion can be found at: http://www.software.ibm.com/data/mediaminer.

Landauer, T., (1987). Research methods in human-computer interaction. In M. Helander (Ed.), Handbook of Human-Computer Interaction. (pp. 905-928). North-Holland, Elsevier Science.

APPENDICES

Appendix 1: Search Participants' Characteristics

Education & Search Experience				
Judgements from 1..5:	None	... Some ...	A great deal	
	S1	S2	S3	S4
education	BA, MA	BA	Ass	Ass
age	under 21	60	25	45
gender	female	male	male	female
previous TREC	no	no	no	no
know test systems	no	no	no	no
online search. exp.	2 yrs.	no	5 yrs.	1 yr.
experience with:				
mouse-based IF exp.	5	3	5	5
comp. library catalogs	3	2	5	1
CD ROM sys.	1	1	5	3
comm. online sys. like Dialog, Lexis	1	1	1	1
web search	5	1	5	4
other systems	2	1		1
full-text DB	2	1	5	2
ranked IR	2	1	1	1
IR w. relev. feedback	2	1	5	2

Appendix 2: Users' Judgements of the Six Experiment Topics

Topics Judgements

Judgements from 1...5: Not at all... Marginally... Extremely

	S1 (ZP)	S2 (IBM)	S3 (ZP)	S4 (IBM)
Topic 303i - 'Hubble Telescope Achievem.'				
familiar topic?	3	3	1	1
difficult search?	2	1	1	4
satisfied w. results?	3	1	5	4
found all aspects?	4	1	4	4
enough time?	5	3	5	4
Topic 307i - 'New Hydroelectr. Projects'				
familiar topic?	4	1	2	1
difficult search?	2	1	1	2
satisfied w. results?	5	1	4	4
found all aspects?	5	1	3	2
enough time?	5	2	5	2
Topic 322i - 'Intern. Art Crime'				
familiar topic?	3	3	4	3
difficult search?	3	1	5	1
satisfied w. results?	4	1	1	5
found all aspects?	2	1	1	5
enough time?	4	1	1	5
Topic 326i - 'Ferry Sinkings'				
familiar topic?	1	1	3	3
difficult search?	3	1	1	3
satisfied w. results?	3	4	3	4
found all aspects?	3	1	3	4
enough time?	4	2	2	4
Topic 339i - 'Alzheimer's Drug Treatm.'				
familiar topic?	3	3	5	4
difficult search?	2	1	1	3
satisfied w. results?	3	1	4	5
found all aspects?	3	1	4	5
enough time?	5	3	4	5
Topic 347i - 'Wildlife Extinction'				
familiar topic?	2	3	5	4
difficult search?	2	4	1	4
satisfied w. results?	5	1	4	2
found all aspects?	5	1	2	2
enough time?	5	3	5	2

Appendix 3: Judgements of Experimental System

IBM's Search System

Judgemt from 1...5:	Not at all...	Marginally...	Extremely	
	S1	S2	S3	S4
easy to use	3	4	5	5
easy to learn	5	4	4	5
understand coverage	4	4	5	5
comments	"stop search" required		required: poss. to re-open docs from saved docs window	a bit slow...

Appendix 4: Judgement of NIST ZPRISE versus Experimental System

IBM Search	**versus**	**ZPRISE**		
Judgement from 1...5:	Not at all...	Marginally...	Completely	
	S1	S2	S3	S4
understand task	5	4	5	5
searches as usual	4	n/a	5	4
different systems	3	no answer	4	4
easier to use	IBM / ZP	ZP	IBM	ZP
easier to learn	ZP	ZP	IBM	ZP
like better	IBM / ZP	IBM	IBM	ZP
comments	ZP easier, since no + / - signs	IBM better look and feel	IBM better look and feel	ZP showed contained query terms

Appendix 5: Search Participants' Detailed Search Data

Subject 1	No. queries	No. q-terms; Av. per query	No. CT refer.s	No. CT usages	No. queries w. operators	No. docs viewed; Av. per query	No. docs saved; Av. per query	Aspectual recall
NQ-326	9	18 (2)	2	1	4	26 (2.88)	4 (0.44)	0.333
NQ-322	13	29 (2.2)	-	-	7	25 (1.9)	3 (0.23)	0.111
NQ-307	3	4 (1.33)	-	-	1	46 (15.33)	10 (3.3)	0.348
ZP-347	6	15 (2.5)	na	na	na	25 (4.16)	11 (1.83)	0.231
ZP-303	4	11 (2.75)	na	na	na	26 (6.5)	3 (0.75)	0.571
ZP-339	7	23 (3.28)	na	na	na	27 (3.85)	4 (0.57)	0.7

Subject 2	No. queries	No. q-terms; Av. per query	No. CT refer.s	No. CT usages	No. queries w. operators	No. docs viewed; Av. per query	No. docs saved; Av. per query	Aspectual recall
ZP-326	2	10 (5)	na	na	na	14 (7)	2 (1)	0.222
ZP-322	3	18 (6)	na	na	na	10 (3.3)	3 (1)	0
ZP-307	1	5 (5)	na	na	na	9 (9)	7 (7)	0.217
NQ-347	14	41 (2.9)	1	5	2	14 (1)	3 (0.214)	0.154
NQ-303	2	5 (2.5)	-	-	1	33 (16.5)	1 (0.5)	0
NQ-339	3	5 (1.66)	1	-	3	9 (3)	4 (1.33)	0.9

Subject 3	No. queries	No. q-terms; Av. per query	No. CT refer.s	No. CT usages	No. queries w. operators	No. docs viewed; Av. per query	No. docs saved; Av. per query	Aspectual recall
NQ-326	3	11 (3.66)	1	1	1	67 (22.33)	4 (1.33)	0.111
NQ-322	11	29 (2.63)	-	-	6	105 (9.54)	3 (0.27)	0.111
NQ-307	1	2 (2)	-	-	-	47 (47)	8 (8)	0.174
ZP-347	3	5 (2.66)	na	na	na	44 (14.66)	4 (1.33)	0.077
ZP-303	1	3 (3)	na	na	na	28 (28)	7 (7)	1
ZP-339	1	3 (3)	na	na	na	30 (30)	5 (5)	0.7

Subject 4	No. queries	No. q-terms; Av. per query	No. CT refer.s	No. CT usages	No. queries w. operators	No. docs viewed; Av. per query	No. docs saved; Av. per query	Aspectual recall
ZP-326	3	21 (7)	na	na	na	26 (8.66)	5 (2.66)	0.333
ZP-322	8	28 (3.5)	na	na	na	25 (3.125)	8 (1)	0.222
ZP-307	2	7 (3.5)	na	na	na	25 (12.5)	10 (5)	0.261
NQ-347	4	13 (3.25)	3	3	-	15 (3.75)	4 (1)	0.038
NQ-303	6	11 (1.83)	2	3	-	23 (3.83)	3 (0.5)	0.286
NQ-339	3	4 (1.33)	2	-	-	21 (7)	4 (1.33)	0.6

Appendix 6: Performance over Search Participant and Search Topic

Average recall per subject:		
	Average NQ recall	Average ZP recall
Subject 1	0.264	0.501
Subject 2	0.351	0.146
Subject 3	0.132	0.425
Subject 4	0.308	0.272
Total	0.288	0.336

Average recall per topic:		
	Average NQ recall	Average ZP recall
326	0.222	0.2775
322	0.111	0.111
307	0.261	0.239
347	0.096	0.154
303	0.143	0.785
339	0.75	0.7
Total	0.263	0.378

Appendix 7: <u>Search Session Report - Topic: Ferry Sinkings</u>

(E = experimenter; S = searcher; SYS = experimental system responses)

E: 'Twenty minutes, here we go. And if you have any questions or comments just say so, all right?'
S: 'Okay.'

types the query: ferry

SYS: displays 50 out of 1083 found documents - [we will show 8 documents here for all result lists]

*** FT922-12800FT 15 APR 92 - Freight ferry
*** FT924-13535FT 15 OCT 92 - New ferry service
*** FT923-3664FT 11 SEP 92 - World News in Brief: Ferry blown up
*** FT934-15680FT 11 OCT 93 - World News in Brief: Fears for Korean ferry
*** FT911-5368FT 15 APR 91 - World News in Brief: Ferries disrupted
*** FT923-9034FT 07 AUG 92 - Ferry row settled
*** FT944-10832FT 09 NOV 94 - SeaCat and ferry collision probed
*** FT924-3846FT 05 DEC 92 - World News in Brief: Ferry blaze

E: (giving support for scanning the result list)
'You are quicker with putting your cursor on the scroll bar.'
S: 'Oh, and just dragging it?'
E: 'Yes.'

S: looks at 2 documents
requests related terms for the query 'ferry'.
P4482 Ferries
O European Ferries
Ex Ferry
P4481 Deep Sea Passenger Transportation

'Now, how do I get rid of these. I just go back up there [into the query entry field], right?'
E: 'Exactly. They will go by themselves. As soon as you press "Documents", it will empty out everything.'

S: changes query to: ferry +sinking +deaths

SYS: displays 50 out of 85 -
***** FT923-4272FT 08 SEP 92 - Arts: Elisabeth
**** FT944-166FT 31 DEC 94 - Swedes count the European costs of bananas and whisky: Hugh Carnegy reports on a nation already unsettled by changes, violence and unemployment as it takes on membership of the EU
**** FT944-6607FT 29 NOV 94 - Technology: Closing in on a serial killer - Cancer kills five people every minute of the day. In the first of a six-part series, Clive Cookson reports on the war against cancer and the encouraging trends
**** FT932-15625FT 10 APR 93 - Sport: The cars are the same, the countries have changed - Motor racing
**** FT911-2716FT 27 APR 91 - Love and death in the city of lost hopes: It is two years since tanks crushed the Peking rebellion. In the heady days before the uprising Peter Ellingsen fell in love with a young student - but the thugs
*** FT922-1708FT 20 JUN 92 - Books: Caught up in family sagas - Fiction
*** FT943-6370FT 30 AUG 94 - Spanish vow to banish drift-nets: Fishing disputes have risen up the diplomatic agenda. FT reporters examine the conflicts worldwide
*** FT943-6371FT 30 AUG 94 - Fleets fight in over-fished waters: Fishing disputes have risen up the diplomatic agenda. FT reporters examine the conflicts worldwide

S: changes query to: ferry +sinking

SYS: displays 50 out of 1383 -
***** FT942-14757FT 19 APR 94 - Letters to the Editor: Channel control overdue
**** FT931-12892FT 27 JAN 93 - Insurers given credit for fall in ships lost
**** FT911-5168FT 16 APR 91 - Threequarters of sunken oil tanker's cargo still on board
*** FT923-2875FT 16 SEP 92 - World News in Brief: Sparring partners
*** FT922-14951FT 03 APR 92 - World News in Brief: Ship's master faulted in Titanic sinking
*** FT923-6450FT 25 AUG 92 - World News in Brief: Malaysia hunts Taiwan trawler
*** FT924-12163FT 22 OCT 92 - Start on N-waste dump planned
*** FT934-14903FT 14 OCT 93 - Observer: Sinking feeling

S: looks at 1 document
'I think I was better with just "ferry".'
changes query back to: ferry

SYS: displays 50 out of 1083 found documents -
*** FT922-12800FT 15 APR 92 - Freight ferry
*** FT924-13535FT 15 OCT 92 - New ferry service
*** FT923-3664FT 11 SEP 92 - World News in Brief: Ferry blown up
*** FT934-15680FT 11 OCT 93 - World News in Brief: Fears for Korean ferry
*** FT911-5368FT 15 APR 91 - World News in Brief: Ferries disrupted
*** FT923-9034FT 07 AUG 92 - Ferry row settled
*** FT944-10832FT 09 NOV 94 - SeaCat and ferry collision probed
*** FT924-3846FT 05 DEC 92 - World News in Brief: Ferry blaze

S: looks at 3 documents

'So, am I supposed to write something about this if I am going to save it?'
E: 'Yes.'
S: 'Like write something about the title?'
E: 'No, write down something that says that this is about the Bangladesh Ferry disaster, that's fine.'
S: 'Okay.'
E: (commenting on UI issue) 'Actually, you only need to click on it (push button) once. I know we had some problems yesterday, like, when you click on it, it may not come back up again.'

S: saves 1 document

'So, I'm only supposed to save one?'
E: 'One per each sinking event with more than 100 dead people - I don't like this topic...'
S: 'So, it doesn't matter. I just have to save as many as I can that have 100 deaths and describe where it is?'
E: 'Yes. Have you ever heard about any ferry sinkings so that you can think of any?'
S: 'No, not really, no idea...'

looks at another 3 documents
saves 1 document
looks at another 2 documents

changes query to: ferry +europe

SYS: .... takes long time ....

E: 'I just know that the search engine is mainly looking for Europe now and...'
S: 'not ferries...'
E: 'among the documents of Europe, also for ferries...'
S: 'So, it would have been better if you wrote Europe plus ferries, then it would have looked for ferries more than Europe?'
E: 'The plus is always the most important.'
S: 'All right, is there any way I can stop this?'
E: 'No, unfortunately not. This Europe was prevalent and it had 36,000 documents, I just happen to know...'
S: 'Well, why wouldn't it do the ferry stuff first? Do you know what I'm saying?'
E: 'Good question.'
S: 'Like if I had put this first...I'll do it that way.'

SYS: displays 50 out 36711 found documents
*** FT941-5746FT 04 MAR 94 - International Company News: Write-off hits Norwegian ferry operator
*** FT922-8897FT 11 MAY 92 - Letter: Right place for locating EuroFed
*** FT911-4515FT 19 APR 91 - Survey of the Canary Islands (8): Green, fertile and volcanic - Western Islands
** FT922-8062FT 15 MAY 92 - Survey: FT Traveller, Genoa (6) - Symbol of lost opportunities - The Harbour, The flag of privatisation is now flying highest at the port - surprisingly, for a former centre of hard-left politics
** FT941-4041FT 12 MAR 94 - Travel: Light fantastic - Why artists colonised a remote Danish fishing village
** FT911-4516FT 19 APR 91 - Survey of the Canary Islands (4): Ports provide that vital economic lifeline - Santa Cruz de Tenerife and Las Palmas de Gran Canaria face competition from Agadir
** FT922-7181FT 20 MAY 92 - Arts: The feel-good factor - Television
** FT921-13774FT 20 JAN 92 - The Week Ahead

S: changes query back to: ferry
and adds a related term to the query: 'O European Ferries'
so query now reads:
ferry 'O European Ferries'

E: 'Take that "O" out of there, it's not going to do anything right now.'
S: 'Why did it put that?'
E: 'These related terms are processed automatically. Many funny terms come from the fact that the Financial Times has specific types of tags that they use.'
S: 'So, I should erase this?'
E: 'Yes.'

[here, the system takes a long time, because of a user / UI error and the intermediate dialog is about this behaviour]

SYS: displays 50 out of 1083 found documents
***** FT941-732FT 29 MAR 94 - Netherlands ferry route may restart
**** FT934-15859FT 09 OCT 93 - P&O announces 240 job cuts
*** FT931-6871FT 26 FEB 93 - P&O negotiating sale of two ferries to Greek shipping companies
*** FT941-3869FT 14 MAR 94 - FT Guide to the Week
*** FT941-10270FT 10 FEB 94 - Management (Marketing and Advertising): Softly, softly approach - The French enlisted history-makers to market the Channel Tunnel, but UK advertising has had to be coaxing and humorous
*** FT941-6094FT 03 MAR 94 - Technology: Check-in for the Channel rail link - A new passenger ticketing system
*** FT922-12800FT 15 APR 92 - Freight ferry
*** FT924-14523FT 09 OCT 92 - Ferguson wins Pounds 16m ferries order

looks at 1 document

S: changes query to: ferry "European Ferries"

SYS: displays 50 out of 1083 found documents -
***** FT941-732FT 29 MAR 94 - Netherlands ferry route may restart
**** FT934-15859FT 09 OCT 93 - P&O announces 240 job cuts

*** FT931-6871FT 26 FEB 93 - P&O negotiating sale of two ferries to Greek shipping companies
*** FT941-3869FT 14 MAR 94 - FT Guide to the Week
*** FT941-10270FT 10 FEB 94 - Management (Marketing and Advertising): Softly, softly approach - The French enlisted history-makers to market the Channel Tunnel, but UK advertising has had to be coaxing and humorous
*** FT941-6094FT 03 MAR 94 - Technology: Check-in for the Channel rail link - A new passenger ticketing system
*** FT922-12800FT 15 APR 92 - Freight ferry
*** FT924-14523FT 09 OCT 92 - Ferguson wins Pounds 16m ferries order

S: looks at 5 documents
saves 1 document
looks at another 3 documents

E: 'Another point is that if you use more than one term, and because we really want to concentrate on ferry now, so put a "+" in front of ferry. Ferry is our basic one. If you only use ferry, then it knows you only want to look for ferry. If you have more than one term, you have to tell them, what is your most important aspect, or topic here.'

S: changes query to: +ferry sinking

SYS: displays 50 out of 1083 found documents -
***** FT944-15057FT 20 OCT 94 - Improved ferry safety urged
***** FT944-5084FT 06 DEC 94 - Ro-ro ferry study agreed
***** FT944-11367FT 07 NOV 94 - Pounds 45m car-ferry research planned
***** FT944-5248FT 05 DEC 94 - Sea safety review focuses on ferries
**** FT944-11048FT 08 NOV 94 - Bow doors faulty on 33% of ferries using UK ports: Government to increase safety checks on vessels
**** FT944-11013FT 08 NOV 94 - International Company News: Heavy loss in US pushes Trygg-Hansa into the red - Swedish insurer posts SKr813m deficit at nine months
*** FT944-10102FT 12 NOV 94 - Tighter ferry rules proposed
*** FT944-10109FT 12 NOV 94 - Tighter ferry rules proposed

S: 'It's hard to find them that say 100 people...a lot of them say "less than 100".'
E: 'That document is an 88 minus...'
S: 'Makes me not want to go on a ferry now.'

[intermediate dialog about a document the searcher thought she had saved, but was not recorded in the saved documents list]

looks at 5 documents

'O, nine hundred!'
E: 'That was a big European event. That was a very, very bad one.'

S: saves 1 document

'So, how many more do I have to look for? I already saved three, am I supposed to look for more than that?'
E: 'I think there are more than that. Now, if you try to do it with the plus ferry and whatever else...'

S: 'Do I have to put the plus though?'
E: 'No.'
S: 'So, I'm not supposed to save them again right?'
E: 'No. So, think of some other means of excluding Estonia. You don't want to see Estonia anymore, because you saved it.'
S: 'Oh, so I put 'minus', right?'
E: 'Right.'

S: changes query to: +ferry sinking -estonia

SYS: displays 50 out of 1029 found documents -
**** FT943-1964FT 21 SEP 94 - Survey of Logistics (7): Flags of inconvenience - Competition flares in Europe's sea-lanes
*** FT922-12800FT 15 APR 92 - Freight ferry
*** FT923-3664FT 11 SEP 92 - World News in Brief: Ferry blown up
*** FT924-13535FT 15 OCT 92 - New ferry service
*** FT934-15680FT 11 OCT 93 - World News in Brief: Fears for Korean ferry
*** FT911-5368FT 15 APR 91 - World News in Brief: Ferries disrupted
*** FT934-1954FT 16 DEC 93 - Technology: Ships bridge the danger gap - Andrew Fisher concludes a series on transport safety with an investigation into innovations that may help prevent sea disasters and give clues to their
*** FT924-3846FT 05 DEC 92 - World News in Brief: Ferry blaze

S: looks at 6 documents

E: 'Do you have the feeling that you'll find any others?'
S: 'They are all the same basically. Like all the ones that have more than 100 deaths seem to be all the same in every area that I put.'
E: 'So, you don't find new instances?'
S: 'No, not with more than 100 people.'
E: 'Well, if you think you are done... - we'll be close to being done here anyhow. This thing (the alarm clock) will ring in a minute.'
S: 'What do I do with the ones that I already saved? Just leave them?'
E: 'Go back, say 'Browse Saved Documents'.'

S: browses saved documents

[again a brief dialog about documents apparently missing from the saved documents list]

Buzzer rings.

E: So, let's stop here.

The GURU System in TREC-6

Eric W. Brown Herb A. Chong
IBM T. J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598
{brown, herbie}@watson.ibm.com

1 Introduction

As the on-line world grows and increases its role in our daily lives, the problems of searching, categorizing, and understanding textual information become ever more important. While researchers and practitioners have made much progress in these areas over the last thirty years, anyone who has gone to the World Wide Web seeking information and returned with more frustration than answers can attest that much work remains. Today's important issues cover topics such as scalability, user interfaces, and techniques that exploit the unique hypermedia features of Web environments.

To support our research in these areas, the Text Analysis and Advanced Search department at the IBM T. J. Watson Research Center has developed an experimental probabilistic text retrieval system called Guru[4]. Guru was originally built to explore new probabilistic ranking algorithms, and now serves as a test-bed for much of our text analysis, search, and categorization work. Guru may be run as a stand-alone system or in a client/server configuration. The Guru indexer performs minimal case and hyphen normalization, but otherwise indexes all words (including stop words) in their original form. The index includes document, paragraph, sentence, and word-in-sentence positional information for each word occurrence in the document collection.

At search time, queries are input to Guru in a free-text format. Stop words are eliminated from the query and morphological variants for each query term are automatically generated and added as synonyms to the query term. Syntax is provided that allows the user to control morphological expansion and stop word elimination. Guru ranks documents using a probabilistic algorithm that considers the frequency statistics of the query terms in individual documents and the collection as a whole. Guru also considers *lexical affinities* (LAs), which are co-occurrences of two terms within a given distance. These automatically identified "phrases" are ranked higher than instances of the component words occurring outside of the LA distance.

Our purpose in participating in TREC-6 is four-fold. First, we continue to refine the base probabilistic ranking algorithm in Guru and wish to evaluate its performance on a large, standard test set. Second, we are developing a prototype user interface and seek initial feedback and guidance for further development. Third, we are interested in text search scalability as an issue orthogonal to the basic problems of search and categorization and seek feedback on initial attempts to address this issue. Fourth, hypermedia domains, such as the World Wide Web, are an increasingly important arena for application of text analysis and search technology. Such domains, however, pose a challenge for evaluation since both search and navigation must be considered by the evaluation metric. We hope that this issue will be addressed by the TREC community with the ultimate goal of defining appropriate evaluation metrics and building suitable test collections. Toward these ends, we are participating in the Ad-hoc Task, the Interactive Track, and the Very Large Corpus Track of TREC-6.

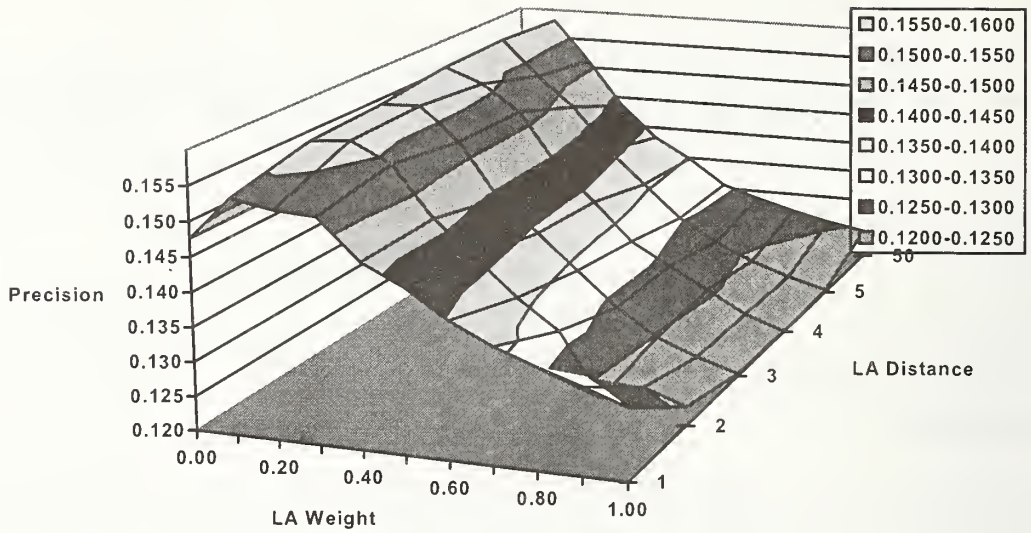


Figure 1: Average precision over varying LA weight and distance

2 Ad-hoc

Our focus in the Ad-hoc Task was to evaluate the performance of our core ranking algorithm. Our TREC-5 results[5] suggested that Guru was using LA scores in a sub-optimal fashion. Most of our pre-submission work involved determining more appropriate settings for the LA distance and the weight of LA contribution to overall document score. A series of experiments was run on the TREC-5 data over which these parameters were varied. Figure 1 shows the average non-interpolated precision obtained by varying LA distance from 1 to 5 and LA weight from 0 to 1. The plot indicates that our probabilistic ranking formula should give LA terms a weight of 0.1 relative to single terms, and the LA distance should be 5. Note, however, that performance is more sensitive to LA weight than LA distance, and the difference in performance between LA distances 1 and 5 is marginal. This is a useful result, since a larger LA distance yields more LA terms for scoring, increasing the processing required to evaluate a query. If similar effectiveness can be obtained with a shorter LA distance, then the system will run faster.

For TREC-6 we submitted two runs, both in the automatic query construction category. One run (ibmg97a) was generated using the topic description field only, while the second run (ibmg97b) was generated from the topic description plus topic title. The script used to generate the queries extracts text from the appropriate topic fields, strips out certain stop phrases and words (based on previous TREC topics), removes punctuation, and produces a query suitable for input to Guru. Most stop words are left in the query at this stage since they are counted in the LA distance when identifying LA terms. Guru ultimately removes stop words from the query using a list of approximately 250 stop words. The queries were run with an LA weight of 0.1 and an LA distance of 5. Note that Guru currently performs no automatic query expansion or relevance feedback.

The results from our two submitted runs are summarized in Table 1. Combining the topic title with the topic description yields a significant improvement over using the topic description alone. A quick analysis of the query topics indicates that a number of the topics (e.g., 308, 311, 312, 316, 328) have significant key words, phrases, or unique morphological variations that appear in the title but not in the description.

Interpolated Precision			
Recall	ibmg97a	ibmg97b	
0.00	0.5709	0.6659	(+16.6)
0.10	0.3557	0.4516	(+27.0)
0.20	0.3080	0.3842	(+24.7)
0.30	0.2578	0.3191	(+23.8)
0.40	0.2180	0.2614	(+19.9)
0.50	0.1716	0.2255	(+31.4)
0.60	0.1331	0.1929	(+44.9)
0.70	0.0642	0.1404	(+118.7)
0.80	0.0283	0.0613	(+116.6)
0.90	0.0086	0.0383	(+345.3)
1.00	0.0071	0.0234	(+229.6)
Ave. prec. (non-interp)	0.1727	0.2309	(+33.7)

Table 1: Ad-hoc automatic category A results (a = short, b = long)

Including these words in the query is usually beneficial. We note, however, that performance deteriorated significantly on at least one query (314) when the title was included in the query. The reason for this has not yet been determined, though the average precision over all TREC-6 participants on topic 314 was worse in the Automatic Long-topics Task than in the Automatic Short-topics Task (Ad-hoc Category A).

Of the 16 participants in the Ad-hoc Automatic Category A Long-topics Task, Guru produced the best average precision (non-interpolated) for 7 of the 50 topics. Guru performed above the median average precision on 29 topics, and below the median average precision on 21 topics. So far we have found no strong correlations between query characteristics and Guru performance, although the topics for which Guru performed best tend to have a relatively small number of relevant documents. This might suggest that inclusion of the topic narrative or the use of automatic query expansion or relevance feedback techniques by other participants tended to reduce their precision. Alternatively, this might suggest that the incorporation of automatic query expansion or relevance feedback techniques into Guru would improve our performance. Of course, we prefer the second hypothesis and plan to prove it in the future.

3 VLC Track

We can attack the execution performance issues associated with large text collections at a variety of levels. Low level techniques, such as Smeaton and van Rijsbergen[7], Buckley and Lewit[2], and Brown[1], use thresholds and constrained candidate document sets to reduce the amount of work performed in the core ranking algorithm. Higher level techniques, such as Stanfill[8], Tomasic and Garcia-Molina[9], and Cahoon and McKinley[3], use parallel or distributed architectures to scale IR system performance. We opted for a high level approach in the VLC Track.

We ran our VLC Track tests using an experimental distributed search system that performs collection fusion across distributed document collections. Our system can distribute queries to an arbitrary number of search servers in parallel and merge the results into a single hit-list. Result merging was performed without rank normalization, working under the assumption that the documents were distributed in such a way that collection wide term statistics were approximately consistent across all search servers. Guru was used as

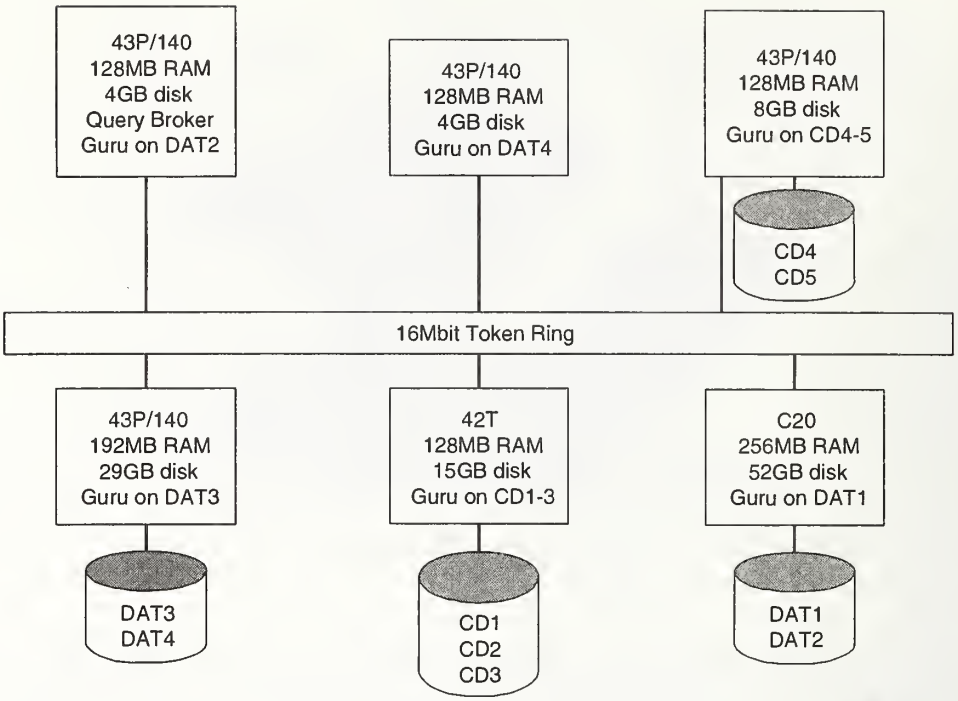


Figure 2: VLC system architecture, showing distribution of data and processing

Collection	Search Servers	Batch Time (min.)	Ave. Prec. @ 20
Baseline (2.02GB)	1	16.5	0.275
Full VLC (17.8GB)	6	47.2	0.361

Table 2: VLC results

the search server in all cases, with an LA weight of 0.1 and an LA distance of 1.

For our VLC runs we used the same queries as were used to produce our Ad-hoc ibmg97a run (automatically generated from topic descriptions only with no automatic feedback or query expansion). Our Baseline run was conducted on a single RS/6000 43P/140. The full VLC run was conducted on a network of six workstations, including four RS/6000 43P/140's, one RS/6000 C20, and one RS/6000 42T connected by a 16Mbit token ring. The data for the full run was stored on only four of the six workstations due to disk space limitations on two of the machines. Figure 2 shows the distributed architecture and distribution of data.

Our VLC results are summarized in Table 2. The batch time reported is the time to process all 50 queries in a single batch. Note that our Full VLC results were obtained on an incomplete VLC collection. Our indexer was unable to parse the long web server log files contained in the AUNI collection on DAT3, causing that collection indexing run to fail. Also, a large portion of the NEWS08 collection on DAT4 failed to unload from our tape and was not indexed. Unfortunately these errors were not detected in time to make the necessary corrections before the runs had to be submitted.

Our results are generally encouraging. Average precision at twenty documents actually improves from the Baseline to the Full VLC. We would like to conclude that this validates our collection fusion approach

with no rank normalization. However, all seven participants in the track experienced a similar effectiveness improvement from Baseline to Full VLC. More analysis is required to determine if this improvement is due more to characteristics of the collections than quality of the retrieval systems.

Our system found an average of seven relevant documents in the top twenty documents returned. This is six fewer than the best performing system in the track, which found an average of thirteen. We note, however, that our queries did not include the topic titles. As discussed in Section 2, adding topic titles to the queries significantly improves performance. Unfortunately, evaluating the results of queries with topic titles now using the relevance judgments produced by the VLC track would be inconclusive since the coverage of those judgments is so small. Without having submitted title plus description runs to the assessors we can only predict, based on our experience in the Ad-hoc task, that our effectiveness would improve.

The execution performance of our system is satisfactory. Under ideal circumstances, we would expect that using 9 machines configured similarly to the baseline machine (each searching data stored on local disk) would allow us to search 18 GB in approximately the same amount of time as required to search the Baseline data (i.e., 16.5 minutes). This ideal architecture was not available, however, forcing us to distribute the Full VLC data such that the largest individual collection searched was over 4 GB, two of the six servers accessed their index data from remote disk, and two of the six servers had to additionally act as file servers. This allows the scalability of a single search server to limit the performance improvement obtainable in the distributed architecture. The Guru search server is a prototype designed for exploring ranking algorithms. As such, execution performance is not a priority. In spite of this, we achieve execution speeds that scale well with collection size. The single machine (Baseline) system requires 9.6 milliseconds/megabyte/query, while the six machine distributed (Full VLC) system requires 3.1 milliseconds/megabyte/query. These results were obtained with the machines in multi-user mode and the runs were made during the evening when the machines and network were lightly loaded.

Our participation in the VLC Track at TREC-6 was motivated by two goals: 1) to obtain preliminary performance data on our prototype distributed search system, and 2) to contribute to the development of a large, realistic test set with meaningful queries and relevance judgments. We have achieved both of these goals to a certain degree. Attempting to index and search 20 GB of data produced valuable feedback on our prototype and identified a number of opportunities for further work. As a participant, we contributed to the track discussions and submitted documents to the judging pool. Ultimately, the relevance judgments for the VLC track appear to have less utility than originally hoped. While the participants can use them to evaluate their submitted runs, the relevance judgments should not be used to evaluate runs that did not contribute to the judging pool. The coverage of the relevance judgments is simply too small, rendering any effectiveness evaluations inconclusive.

4 Interactive Track

Our participation in the Interactive Track is described in detail in Schmidt-Wesche *et al.*[6] To conduct our Interactive Track experiments, a number of search and text analysis systems were used, including IBM's NetQuestion search engine and Guru. NetQuestion is a hybrid Boolean and probabilistic free-text ranking system. It uses a version of the Guru ranking algorithm to perform probabilistic free-text ranking.

5 Summary

This is the second year in which the Text Analysis and Advanced Search department at the IBM T. J. Watson Research Center has participated in the TREC conference. For TREC-6 we have expanded our participation

beyond the Ad-hoc task to include the Interactive and VLC tracks. While we continue to improve our core ranking algorithm, our efforts this year were focused mainly on the Interactive and VLC tracks. In the future, we hope to see TREC activity in the hypermedia domain. Information retrieval in a hypermedia environment is a unique combination of search and navigation. The creation of appropriate hypermedia test collections and metrics would be of great value to this research community.

Acknowledgements

Thanks go to Paul Jensen for helping prepare the VLC data.

References

[1] E. W. Brown. Fast evaluation of structured queries for information retrieval. In *Proc. of the 18th Inter. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 30–38, Seattle, WA, July 1995.

[2] C. Buckley and A. F. Lewit. Optimization of inverted vector searches. In *Proc. of the 8th Inter. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 97–110, June 1985.

[3] B. Cahoon and K. McKinley. Performance evaluation of a distributed architecture for information retrieval. In *Proc. of the 19th Inter. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 110–118, Aug. 1996.

[4] Y. S. Maarek and F. A. Smadja. Full text indexing based on lexical relations. In *Proc. of the 12th Inter. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 198–206, Cambridge, MA, June 1989.

[5] Y. Ravin. The GURU system in TREC-5. In D. Harman and E. Voorhees, editors, *The Fifth Text REtrieval Conference (TREC-5)*, Gaithersburg, MD, 1997. National Institute of Standards and Technology Special Publication 500-238.

[6] B. Schmidt-Wesche, R. Mack, C. Cesar, and D. VanEsselstyn. IBM search UI prototype evaluation at the interactive track of TREC6. In D. Harman and E. Voorhees, editors, *The Sixth Text REtrieval Conference (TREC-6)*, Gaithersburg, MD, 1998. National Institute of Standards and Technology Special Publication.

[7] A. F. Smeaton and C. J. van Rijsbergen. The nearest neighbour problem in information retrieval. An algorithm using upperbounds. In *Proc. of the 4th Inter. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 83–87, Oakland, CA, 1981.

[8] C. Stanfill. Parallel computing for information retrieval: Recent developments. Technical Report TR-69 DR88-1, Thinking Machines Corporation, Cambridge, MA, Jan. 1988.

[9] A. Tomasic and H. Garcia-Molina. Performance of inverted indices in distributed text document retrieval systems. Technical Report STAN-CS-92-1434, Stanford University Department of Computer Science, 1992.

Concrete Queries in Specialized Domains:

Known Item as Feedback for Query Formulation

Mun-Kew Leong
Institute of Systems Science
Singapore

Abstract

This paper investigates the use of short concrete queries as being typical of non-naive non-computer professionals in information searching and retrieval. We aim to simulate such an environment with a general corpus such as the TREC data using what we term "known item query formulation". Assuming the simulation is valid, the results suggest that such searching will satisfy the profiled user.

1 Introduction

We introduce the concept of a *concrete query* as opposed to an *abstract query*. We will provide a formal definition of both later. Intuitively, however, concrete queries may be thought of as those which are based on examplars, and abstract queries as those which are based on descriptions.[1] In some cases, queries can be *mixed*, i.e., a combination of the two.

1.1 Motivation

We approached the TREC experiments this year with an interest in the methodology of searching in specialized domains. We conducted informal discussions with non-computer professionals who, however, were well versed in using a computer, and moreover, had some to considerable experience searching for information on-line. We had the intuition that such searchers would use short queries with highly specialized terms, i.e., what we have described above as concrete queries.

1.1.1 Interviews

We talked separately to the following non-computer professionals:

1. a psychiatrist with more than five yrs experience using computers and who had been searching on-line and on the WWW for over three years. On-line searching experience included OPACs, Medline, etc.
2. a management consultant with a degree in Economics who had been using computers for over ten years, and who had been searching the internet for over 4 years. On-line searching experience are on business databases and the WWW.
3. a doctor with two years computer experience, and the same amount of time searching the WWW.

[1] This distinction has been otherwise termed, respectively, *referential* and *descriptive*.

The aim of the discussion was to elicit the form of the queries which these users employed to search their own areas of expertise on-line, and to get an idea from them if the way they searched on-line was similar to how their colleagues did.

While this survey was admittedly informal, and there were only three persons polled, the following information was noted:

- the users mainly used keyword (non-Boolean) search
- the query terms used depended on the search. Names of procedures, drugs, people, companies, etc., were often used. Descriptions of the things to be searched were rarely written out before hand, except as reminders.[2]
- all three users knew of, and sometimes used, and's and or's in their queries. None of them used not's. If they used Boolean operators, it was usually and's, and only when there were too many records found, and they could not find relevant records/documents among them.
- the users all revised their queries based on results of previous searches.
- queries varied in length. Initial queries were short, but revised queries could be quite long
- only one of them knew about phrase operators and mandatory operators on WWW search engines.
- all of them were satisfied with their ability to search. They thought that they could find the information they looked for most of the time.
- none of them knew how their colleagues searched. They thought it would be similar.

While it may not be clear in the observations above, all three users agreed that short queries using names and a few specialized terms would give good results.

1.2 Aim

We therefore conceived an experiment with the following aims:

1. determine if concrete queries would actually give good results
2. determine if manually constructed concrete queries would perform better than relevance feedback using a document which contained the same concrete search terms
3. determine if we could meaningfully use the TREC structure to simulate testing of concrete queries. This would allow us to continue experiments with concrete queries while taking advantage of the TREC topics and relevance judgements.

2 Concrete and Abstract

Taking an example from the TREC topic 339, an abstract query would resemble:

> drugs being used in the treatment of Alzheimer's Disease

whereas a concrete query would be something like the following:

[2]Interestingly, the reminders were not descriptions of the information to be looked for, but rather the problem to which the information was to be applied.

tacrine hydergine velnacrine selegiline THA cognex mentane

Obviously, the latter (concrete) query requires a much more specialized knowledge of the domain queried. That, however, is what we believe are the types of queries used by sophisticated searchers (doctors, scientists, etc.) and which we are interested in exploring further.

2.1 Formal Definitions

This section lays a preliminary foundation for formal work on concrete and abstract representations, but a more complete exposition is beyond the scope of this paper.

We define a *term* as a non-empty sequence of characters (in any orthographic system, e.g., the Roman alphabet, or the Chinese character set).

We define a *representation scheme* as the set of terms used to index or represent a domain (corpus) of objects (documents). For example, a representation scheme could be an inverted word index or the set of features in a vector model.

Formally, given a domain of objects to be represented, a representation scheme is *concrete* if it partitions the domain into **non-overlapping** classes of objects. For example, if a set of documents was partitioned by the terms "round" and "square", no document could be in both partitions since there are no round square objects. The Vienna Classification for trademarks would be an example of a concrete representation scheme.

Conversely, the representation scheme is *abstract* if it partitions the domain into **possibly overlapping** classes of objects. For example, suppose an indexing scheme partitions a corpus of documents based on the two terms "blue boxes" and "rectangular boxes", and there are documents which talk about rectangular blue boxes, and are thus members of both classes, then that indexing scheme (or representation scheme) is abstract.

We define a term as *concrete* if it is an element in a concrete representation scheme, and *abstract* if it is an element in an abstract representation scheme. We can informally think of *concrete* and *abstract* as properties of terms used to index or represent objects (documents) over a given domain (corpus).

In reality, however, a corpus of documents would have both a concrete and an abstract representation scheme. The terms used in each scheme are, unfortunately, indistinguishable in and of themselves (as strings of characters). Nevertheless, relative to a given corpus, we can still think of terms as being abstract or concrete.

3 The Experiment: Modelling Concrete Queries

We attempt to emulate this scenario in the generalized TREC environment using what we term "known item query formulation". This is a manual 2-step query formulation method.

3.1 Terminology

The following terms are used in the description of the experiment, which may require explanation:

- **highly relevant:** there is no distinction in degree of relevance in the TREC relevance judgements, but the subject was instructed to use a document (or part of a document) which addressed the topic and which contained concrete terms such as names (of people, places, procedures), objects, etc. Such a document was termed *highly relevant.* A document is *relevant* if it addressed the topic but did not contain concrete terms.
- **topic:** a topic is the TREC topic.
- **query:** a query is an instantation of the TREC topic as fed to the search engine.
- **iterative construction:** a query is constructed and fed into the search engine. The subject makes use of the results of the query to amend his or her earlier query. This is done until the query satisfies the required stop condition, or the subject runs out of time.

3.2 Method

The latest TREC ad-hoc topics (topics 301 to 350) were used to establish the search task for the subject. For each TREC topic,

1. the subject was given 20 minutes to find one or more documents deemed highly relevant (see above) in the corpus. Tools used were grep and its variants. There was no access to the search engine in this step.

   It is possible that the subject not find a highly relevant or even a relevant document within the time limit.

   The subject was asked to keep note of the documents deemed (highly) relevant. If none were found, then this was to be stated.

2. the subject was then given 10 minutes to iteratively construct a query which returns at least one of the documents found in Step 1 in the top 10 ranked documents using the search engine. The subject was instructed to construct short queries if possible, and to stop once the condition was satisfied.

   This query is saved and used for the run. No query expansion or pseudo-relevance feedback was employed.

   It is possible that the subject not be able to construct a query which returns any relevant documents in the top 10 ranked list.

   In the case that the subject not find any relevant documents in Step 1, the subject was asked to construct what he or she thought to be the best query.

Step 1 basically creates the scenario that the subject has a "resource" (in this case the documents deemed relevant, in the true case, their domain expertise) from which they may mine relevant concrete terms from which to construct their query. The time limit was imposed purely to make the experiment manageble.

Step 1 may also be thought of as identifying a *known item* for retrieval. Once that item is found in the top 10 documents in Step 2, the query is fixed. This was what was meant by the sub-title "known item as feedback for query formulation".

Just to be clear, in Step 2, the subject has access to all and only the following in the construction of his or her query:

- the original TREC topic statement in its entirety
- documents found in Step 1 which are deemed highly relevant, or which are deemed relevant.
- documents returned by the search engine in Step 2, their order, and their contents.
- any personal knowledge with which to construct or augment the queries

4 Running the Experiment

4.1 Subject and Experimental Conditions

There are 50 TREC topics, and with half an hour for each topic, this is a fairly fatiguing and time consuming experiment. Thus, so as not to stress the subject (with consequent effects on performance), and to minimize fatigue, the experiment was conducted over the course of one week. Because of time constraints, only one subject was used. Estimated total time of the experiment was 22 hours. Actual time spent logged on to the workstation was just over 18 hours.

The subject was a staff of ISS, with about 10 years of computer experience, and was well versed in using a UNIX workstation, and with using UNIX system and search tools such as grep, fgrep, cat, more, etc. The subject was allowed to use (and did use) multiple windows, running several searches in parallel.

4.1.1 Problem with Relevant Documents

There was a problem in the experiment. The subject did not keep a list of the documents deemed relevant in Step 1 for each topic. A list was reconstructed by the subject the following week listing at least the topics for which relevant documents were found, and for which relevant documents were not found. Where relevant documents were found, the subject (using the search engine) tried to identify what he had used as relevant and highly relevant documents for each topic. He is sure there are missing relevant documents in his list, but is quite sure that no documents were falsely identified as being relevant. Recall again that "relevant" in this context is with respect to Step 1 of the experiment, and not with respect to TREC relevance judgements.

4.2 Example queries

We examine some of the constructed queries, where relevant documents were found in Step 1 and where no relevant documents were found. We will later analyse the results along the same lines.

4.2.1 Topics with relevant documents

Out of the 50 topics used in the experiment, the subject found highly relevant documents for 29 of them in Step 1. The following two examples particularly show the addition of concrete terms.

- **Topic 331:** World Bank Criticism

Desc: What criticisms have been made of World Bank policies, activities or personnel
Narr: This query is looking for any instances where the World Bank has been accused of things like not being responsive to the unique problems of individual countries, of being too strict in its policies, of pursuing agendas that are biased because of their benefits to western countries, of being no longer useful or practical, of its personnel being difficult to work with, etc.

Constructed Query: world bank criticism criticize environment funding damage lending projects barber conable lewis preston

- **Topic 339:** Alzheimer's Drug Treatment

  Desc: What drugs are being used in the treatment of Alzheimer's Disease and how successful are they?
  Narr: A relevant document should name a drug used in the treatment of Alzheimer's Disease and also its manufacturer, and should give some indication of the drug's success or failure.

  Constructed Query: alzheimer tacrine hydergine velnacrine selegiline THA cognex mentane

However, other queries where highly relevant documents were found did not seem to be very concrete:

- **Topic 307:** New Hydroelectric Projects

  Constructed Query: new hydroelectric projects construction

- **Topic 312:** Hydroponics

  Constructed Query: hydroponics

As it turns out, these simple constructions were sufficient to retrieve the relevant documents (from Step 1) in the top 10, and so the query was frozen at that point.

4.2.2 Topics with no relevant documents

The 21 topics with no relevant documents found in Step 1 were examined and none of the constructed queries had concrete terms. Examples, however, varied widely, with some examples as follows:

- **Topic 335:** Adoptive Biological Parents

  Constructed Query: stepparents stepfather stepmother stepchildren stepfamilies biological parent court decision

- **Topic 319:** New Fuel Sources

  Constructed Query: fuel research

4.2.3 Query Length

One of the issues with query formulation has been query length. The average query length reported for Web search engines[3] is between two and three words long. The average length of the TREC topics, using only the Title and Description fields is 20.2 words long.

Based on the queries constructed above,

- the mean length of all the queries was 7.1 words
- the mean length of queries where relevant documents were found was 5.7 words
- the mean length of queries where no relevant documents were found was 4.2 words.

As we said in our observations in Section 1.1.1, queries varied in length, and started out short, but revised queries could be quite long. This is not disproved by the above evidence.

5 Retrieval Results

We present the general results from running the ad-hoc task both to situate the engine with respect to other systems, and to provide baselines for interpretation. Again, we stress that no pseudo-relevance feedback was employed. In our engine, this typically (based on previous TREC topics and data sets) increases average precision by about 10%.

5.1 General results

We are interested in two of the submitted ad-hoc runs, that of *iss97man* which is the manual query formulation (i.e., the experiment), and *iss97s* which is the baseline formed from the short query task, i.e., the ad-hoc task run using only the Title and Description fields from the topics. These results are shown in Figure 1. As expected, manual query

	iss97man	iss97s
Recall/4611	2792	1599
Precision @0.2	0.4512	0.1889
Precision @0.5	0.2517	0.1206
Average Precision	0.2576	0.1135

Figure 1: Ad-hoc results for *iss97man* manual query formulation

formulation produces much better results than the automatic short queries, especially with no pseudo-relevance feedback.

Incidentally, to provide perspective, we compare the performance of the manual run, *iss97man* relative to all the manual runs submitted for TREC-6. For relevant documents retrieved, *iss97man* performed $\geq$ median on 42 of the 50 topics, with 8 matching the best

[3]Doug Cutting, Excite, during *Panel Session on "Real World" Information Retrieval*, SIGIR'97

results for a topic. For average precision, it performed $\geq$ median on 35 of the topics, however with only 1 matching the best results for a topic.

5.2 Investigation results

The number of topics in which highly relevant documents (in the sense used in this experiment) were found was 29. This means there were 21 documents where no such documents were found. The topics can therefore be divided into two classes: with-relevant and without-relevant. Since all the queries were constructed by the same subject, the without-relevant class is suitable as the *control* topics. The with-relevant class is the *experimental* topics.

The actual results for the two classes are shown in Figure 2. Since the number of topics

	experimental (29 topics)	control (21 topics)
Recall	73% (1759/2401)	47% (1033/2210)
Precision @0.2	0.5348	0.3357
Precision @0.5	0.3238	0.1520
Average Precision	0.3221	0.1685

Figure 2: Raw Control and Experimental results

in each class is different, where appropriate (as for the Recall figures), comparisons between the control and experimental topics will be done on a percentage basis. The difference between the two are obviously significant.

Comparing the performance of the control and experimental topics relative to the median results of all the manual runs submitted for TREC-6, we have Figure 3 for Recall performance, and Figure 4 for Average Precision peformance. We include percentage figures for easier comparisons. Numbers do not add to 100% because "matching top" figures are included in "median or better" figures.

	experimental (29 topics)	control (21 topics)
matching top	7 (24%)	1 (5%)
median or better	25 (86%)	17 (81%)
less than median	4 (14%)	4 (19%)

Figure 3: Recall median performance for Control and Experimental Topics

Looking at Figure 3, we see that recall figures are not particularly different, a 5% greater "median or better" topics for the experimental topics not being significant with such a small sample.

Looking at Figure 4, we see that the average precision figures are very different. Both the "median or better" figure and the "less than median" differ in consistent (i.e., opposite)

	experimental (29 topics)	control (21 topics)
matching top	1 (3%)	0 (0%)
median or better	24 (83%)	11 (52%)
less than median	5 (17%)	10 (48%)

Figure 4: Precision median performance for Control and Experimental Topics

directions by about 30% in both cases. This is significant.

6 Conclusion

We look at Section 1.2 where we detailed our experimental motivation. We have confirmed that concrete queries do give better results, specifically with higher precision. Due to lack of time, we were unable to confirm if manually constructed concrete queries perform better than relevance feedback using documents containing the same concrete search terms. We have also determined that it is possible to use the TREC structure to simulate testing of concrete queries.

As we said, the evidence supports our intuition that expert users tend to retrieve more precise results. We believe this is based on their specialized and concrete vocabulary, but this will have to be investigated further.

While it is good to have the hypothesis experimentally confirmed, even with such a small sample, there is really nothing particularly unexpected about the results. There are, however, two other interesting thrusts arising from this experiment. The first is the formal foundational approach in Section 2, which needs to be developed further. The second is whether the experimental procedure outlined in this experiment can be automated, i.e., if known item feedback can be automatically invoked in non-interactive query formulation.

Preliminary Qualitative Analysis of Segmented vs Bigram Indexing in Chinese

Mun-Kew Leong, Hong Zhou
Institute of Systems Science
Singapore

Abstract

This paper investigates merging multiple methods of indexing for Chinese IR. Identical queries, differently segmented, are used to retrieve individual lists of documents which are then merged before evaluation. Two simple merge methods are discussed. Results on Chinese TREC queries 1 to 28 show improvement over either one of the indexing schemes by themselves. In addition, we examine the difference in the documents returned by each indexing method, i.e., do different indexing schemes retrieve different documents, or the same documents ranked differently, or something else. While we contrasted bigram based indexing with segmented based indexing, the same methods would apply between any two forms of indexing.

1 Introduction

Different queries, different search engines and different indexing schemes give different result sets for any given topic. We expect these results to differ, but it has not been investigated just *how* they differ. Specifically, we might ask the following questions:

- are there trends identifiable when holding one (or two) of query, engine, and indexing method constant, and varying the remainder?
- if there are trends, or correlations, do they identify with any intuitive (cognitive) classes?
- how consistent are these differences? For example, chinese trigram indexing consistently works well when searching for (Chinese) names, but on the average (based on the Chinese TREC queries) perform at about 60% of bigram indexing.

Lastly, we need also be concerned with what kinds of methodology and tools are needed and which are available for doing investigations into this area.

This TREC experiment was aimed at addressing some of those questions above. We focused on short queries using only the Chinese "Desc" field, and compared the result sets obtained by keeping query and engine the same, and varying the indexing method. Specifically, we looked at bigram vs. segmented indexing for the Chinese ad-hoc task. We also investigated various methods for merging the differing result sets to see their effects, and to try and account for differences, if any.

In preliminary work, using the TREC-5 queries, we obtained modest to significant improvement in recall and precision when we merged the results of individual bigram and segmented indexing runs. In both normalized and unnormalized fusion experiments, the merged results were never significantly worse than either individually over a range of query

lengths, types of n-gram indexing, and two types of segmentation. In some cases, we had significant improvements, as will be described later.

2 Outline of Procedure

The experiment had the following basic outline.

1. create two indices, I_1, I_2, corresponding to two different indexing methods (e.g., bigrams and segmented (or "word" based) approches).
2. for each query, q_i, in a given set of queries,
   (a) run q_i against I_1 and I_2 to give lists α_i and β_i, each of n ranked documents.
   (b) extract the following: $\alpha_i \cup \beta_i, \alpha_i \backslash \beta_i, \beta_i \backslash \alpha_i, \alpha_i \cap \beta_i$. These are each sets of results showing some relationship between the two indexing methods for that query.
   (c) take top n documents in each list, and run against relevance judgements for that query
3. merge the results for all the queries, and run against all the relevance judgements to get comparative precison and recall figures to compare against baselines, etc., and to get an overall idea of relative performance of the two indexing methods under investigation.
4. analyse data for the two steps above

In our case specifically, the engine was the same for both indexing schemes, and the queries generated from each topic were the same, except insofar as the tokenization for bigram or segmented querying was necessary.

2.1 Variables

While simple, the methodology above allows us to investigate quite a few different variables, including the following:

- types of indexing:
  - segmentation (compound words, phrases)
    - greedy, short algorithms
    - dictionaries, types of dictionaries
  - n-grams (bigrams, unigrams, trigrams)
- merging returned records:
  - using raw scores
  - using normalized scores
    - within index
    - w.r.t. query terms (no., size)

A slight variation in the outline would also allow us to investigate differences in queries generated from the same topics. However, this would require that we could reliably qualitatively differentiate the differences in queries in a non-statistical manner.

2.2 Data Available

There is also a wealth of data to be mined from the statistics obtainable above. For example:

- Data:
  - S = segmented document result list
    B = bigrams document result list
    $n = 100$, $\cup$ = simple merge based on raw scores
  - $| S \cup B | = 140 \Rightarrow | S \cap B | = 60$ (i.e., 60 common documents between the lists)
  - Consider $T = S \cup B \mid 100$ (the list restricted to the top 100 docs)
  - $| T_S | = | T \backslash B | = 25$, $| T_B | = | T \backslash S | = 75$
- Relevance judgements on:
  - S, B, $S \cup B$, T, $T \backslash S$, $T \backslash B$, $S \backslash T$, $B \backslash T$, etc.
- Series of $n = 100, 200, 300, \ldots$
  - avoid floor and ceiling effects

2.3 Data Analysis

Given the data above, what are the interesting things to look for?

- What can the data tell us?
  - do different indices imply different documents retrieved?
  - do different indices imply different *relevant* documents retrieved?
  - how do the different indices relate with respect to precision vs. recall?
  - at what point do they begin to differ?
  - can we get both (high precision and high recall) with judicious fusion?
  - consider also the distribution of relevant documents
- Look to the document content for "why"
  - intuitive classes of documents?
  - consistency?
  - trends?

3 Results from TREC-5 Queries

The following results (Figure 1) were obtained from Chinese queries 1 TO 28 using bigrams in one case, and a dictionary based segmentation method in the other. The dictionary used was the one compiled by the University of Berkeley, and a greedy (longest match) algorithm was employed. The merging algorithm was to use the maximum score for intersecting documents and the original weight returned otherwise. No corrections were done for the different IDF values in the two indices.

Indexing	Segmented	Bigram	Merged
Recall/2182	1959	2119	2124
Precision @0.1	0.5125	0.5760	0.5724
Precision @0.5	0.3504	0.3864	0.3881
Avg Precision	0.3328	0.3683	0.3686
Exact R-Precision	0.3602	0.3998	0.3975

Figure 1: Various Indexing approaches, ad-hoc task using TREC-5 Data

Switching to a normalized merging algorithm (Figure 2), where scores were normalized with respect to the average score for a query returned by the respective indexing method, we get the following improvement:

Indexing	Segmented	Bigram	Merged
Recall/2182	1959	2119	2125
Precision @0.1	0.5125	0.5760	0.5776
Precision @0.5	0.3504	0.3864	0.3960
Avg Precision	0.3328	0.3683	0.3802
Exact R-Precision	0.3602	0.3998	0.4101

Figure 2: Improved merging algorithm, ad-hoc task using TREC-5 Data

The improvement is approximately 3% and is consistent across the various measures.

3.1 Examples of Specific Queries

We can also look at the statistics obtained from each of the queries. This gives us a better picture of which query was helped, hindered, or unchanged by the merging.

3.1.1 Topic 1

For example, consider just the top 50 records from topic 1. As we can see from the table below (Figure 3), looking at the fine grained level of just the top 50 records can give us some interesting insights.

Indexing	Sgmt	Bigm	Union	Inter	$S \backslash B$	$B \backslash S$
Cardinlty	50	50	85	15	35	35
Relvnt/13	3	9	9	3	0	6
Avg Prec	0.069	0.295	0.262	0.173	0	0.094
Prec 10	0.200	0.300	0.300	0.200	0	0.100
Prec 20	0.150	0.150	0.150	0.150	0	0.200
Prec R	0.154	0.231	0.231	0.231	0	0.077

Figure 3: Topic 1, top 50 documents, various data

As we can see, every relevant document found by the segmented method was also found by the bigram method. If we follow this query for the next 50 records, and then the next, etc., we will find the point (If ever) that the segmented method might find a record not found by the bigram method.

3.1.2 Topic 11

Similar to above, but for topic 11.

Indexing	Sgmt	Bigm	Union	Inter	$S \backslash B$	$B \backslash S$
Cardinlty	50	50	77	23	27	27
Relvnt/186	21	10	24	7	14	3
Avg Prec	0.042	0.011	0.031	0.012	0.038	0.006
Prec 10	0.200	0.200	0.200	0.300	0.500	0.200
Prec 20	0.300	0.200	0.200	0.350	0.500	0.150
Prec R	0.113	0.054	0.129	0.038	0.075	0.016

Figure 4: Topic 11, top 50 documents, various data

In this case, we see that the segmented approach works better, but there are 3 documents retrieved (in the top 50) by the bigram approach that are not retrieved by the segmented.

Again, further investigation with the next 50 results, etc., will give more information.

4 Results from TREC-6 Queries

Given the positive results from the TREC-5 topics, we decided to investigate the relative contributions of the two indexing methods in TREC-6, e.g., what kind of overlap was there in the documents retrieved on the same query by the different methods, where records retrieved by only one method ranked, and where *relevant* records retrieved by only one method ranked. We avoided using pseudo-relevance feedback even though that has significant improvement in earlier experiments (including the TREC-5 experiments above) to minimize the variables involved. The merging method used was to merge based on the raw score, removing the lower scoring duplicate documents.

The following table summarizes the results obtained in the submitted runs for TREC-6:

Indexing	Segmented	Bigram	Merged
Recall/2958	2619	2802	2723
Precision @0.1	0.7686	0.8103	0.7950
Precision @0.5	0.4964	0.6037	0.5095
Avg Precision	0.4709	0.5646	0.4903
Exact R-Precision	0.4689	0.5515	0.4941

Figure 5: Various Indexing approaches, ad-hoc task using TREC-6 Data

The results here go opposite from what occured with the TREC-5 topics. In all the measures above, the merged results are worse than the bigram results. However there is also consistency: the segmented approach was worst in all cases, the bigram was best in all cases, and the merged approach was between them. The merged numbers are closer to the segmented numbers, which may indicate that the merging algorithm favoured the segmented approach.

In looking at the results, this turned out to be the case. Changing the merging algorithm to the average based one used for TREC-5 queries in the previous section brought the merged numbers much closer to the bigram numbers, but bigrams still turn out to perform better.

There is some possibility that a ceiling effect may be in operation. Looking at overall statistics, the bigram approach had 18 $\geq$ median and 4 best at the 100 relevant retrieved level, 24 $\geq$ median and 16 best at the 1000 relevant retrieved level, and 19 $\geq$ median and 1 best with respect to average precision.

Note also that the above results were obtained without the use of pseudo-relevance feedback. When used on the TREC-5 queries, this feedback generally improved precision and recall by about 10%.

Unfortunately, due to lack of time, finer grain analysis (i.e., at the 50 document level, etc.) demonstrated for TREC-5 queries in the previous section was not done on the TREC-6

results. This would greatly help towards understanding how merging changes the documents retrieved and their ranking with the number of documents considered, and would avoid any ceiling or floor effects.

5 Conclusions

There are two sets of conclusions to draw here. With respect to the results obtained from the TREC-6 Chinese track, we found the following:

- TREC-6 results indicate bigrams still perform best
  - documents returned by other indexing methods are mostly subsets of bigram documents
  - segmentation is better only in isolated cases
- so far, no clear cognitive or linguistic classes are identifiable
- segmentation rankings and bigram rankings have different shapes, implying that simple merging algorithms may not work well.

With respect to the methodology proposed, however, it is obvious that such methods generalize between any two indexing schemes. We believe that the data gathered and the possible analyses to be done can shed a lot of light on the qualitative differences between different forms of indexing.

We hope to continue the analysis started on the TREC-6 data, and also intend to apply similar analysis between various n-grams to further understand the relationship between n-gram and segmented indexing. There is some anecdotal evidence that suggests that trigrams would give better precision, and this, among other hypotheses, will be tested in future work.

Experiments on Proximity Based Chinese Text Retrieval in TREC 6

K. Rajaraman[1], Kok F. Lai and Y. Changwen
Information Technology Institute
11, Science Park Road, Singapore 117685.

Abstract

In TREC 6, we participate in the Chinese track and report our experiments on proximity based text retrieval. Our participation this year concentrates on automatic retrieval methods natural for the Chinese language.

We index the documents by treating every Chinese character as a single term and store positional information for all terms. During retrieval we employ a proximity operator that uses the positional information in the index, to rank the documents. The operator is defined such that documents are scored in proportion to the proximity of characters as they appear in the query. Since we only use the proximity of characters to compute the score, the algorithm does not strictly require the word boundaries be known a priori. In particular, phrase detection can be derived as a special case of our algorithm by giving maximum score when the characters are immediately adjacent and 0 otherwise. This indexing and retrieval scheme is significantly different from our TREC 5 method.

We submit three official runs itich1, itich2 and itich3 for TREC 6. For itich3, we use all phrases from the Description field and compute scores with our proximity operator. The runs itich1 and itich2 are obtained through automatic query expansion methods. We dynamically build a 3-gram phrase dictionary from top 20 documents for each query ranked in itich3 and pick phrases to expand from this dictionary using document frequency estimates. The run itich2 is different from itich1 in that the expanded phrases are filtered to remove duplicate and common phrases.

1 Introduction

In TREC 5, we participated in Routing, Filtering and Chinese tracks[NL96]. Due to time and space constraints, this year we take part in the Chinese track only. Our last year experiments include both automatic and manual Chinese text retrieval. Our experiments in TREC 6 concentrate on automatic methods natural for the Chinese language.

It is well known that Chinese text is written as a string of ideograms with no specific word delimitter as in languages like English. Words are identified based on the context in

[1]Please direct all enquiries and correspondence to K. Rajaraman, Information Technology Institute, 11, Science Park Road, Singapore 117685. Email : kanagasa@iti.gov.sg

which they appear. This distinguishing feature of Chinese language (and a few other Asian languages) demands a departure from conventional methods for Chinese IR.

Broadly there are two approaches to Chinese text retrieval *viz.* linguistic and non-linguistic. Of these, the non-linguistic approach is the best investigated one for large applications[BSM96, ACC+96, KG96, BGH+96, GCH+96]. This approach is based on representing Chinese in an English like structure and applying conventional IR techniques for English. Typically, a word segmenter (eg. longest match algorithm. See, for instance, [BGH+96]) is applied on the text collection and the segmented words used to build the index. During retrieval, the query is segmented in a similar way and the resulting words matched with the index to score the documents. Stopword removal and weighting schemes like *tf.idf* are employed to improve precision and recall. As it can be observed, this is essentially a process of mimicing English IR for Chinese by using segmented words as index terms. This method crucially depends on the accuracy of the segmentation algorithm. For instance, if the segmentation algorithm returns the longest word but the query contains only a short word, then the result may be a partial match. This is an inherent problem with word-based indexing. Character based indexing is a more flexible method. In character based indexing, either 1-grams, 2-grams or 3-grams can be used as index terms. 1-gram terms are ambiguous and they may adversely affect precision. 2-grams and 3-grams carry more specific meaning than 1-grams and are usually adequate as index terms[LA96, Lin72]. However, the size of the 2-grams index may be too large to be manageable. Even for a collection with 7000 characters (in GB set), theoretically there could be 49 million 2-gram terms. Hence, to tackle the retrieval problem effectively, we need a novel approach more natural for the Chinese language. Our TREC 6 work is a step in this direction.

2 Proximity Based Text Retrieval

Our approach is based on ranking documents using the proximity of characters in the query string.

We index every character in the document collection and store the positional information of all occurences of the terms. The positional information will be used in scoring the documents as below.

Suppose $c_1c_2....c_n$ is a Chinese string. We define a proximity operator

$$Prox_i(c_1c_2...c_n) = \sum_{k=1}^{n-1} \frac{f\left(Dist_i(c_k, c_{k+1})\right)}{n-1}$$

where i stands for the document number and

$$Dist_i(c_k, c_{k+1}) = \text{smallest positive distance from } c_k \text{ to } c_{k+1} \text{ in } i\text{-th document}$$

i.e. If we define

$$pos_{ij}(c_k) = \begin{cases} \text{the position of } j\text{-th occurrence} & \text{if } c_k \text{ occurs at least } j \text{ times} \\ \text{of character } c_k \text{ in } i\text{-th document,} & \text{in the } i\text{-th document} \\ 0, & \text{otherwise} \end{cases}$$

then

$$Dist_i(c_k, c_{k+1}) = min_{j,l}(max(0, pos_{ij}(c_{k+1}) - pos_{il}(c_k)))$$

The function $f : R \rightarrow [0, 1]$ (called the *proximity* function), is non-decreasing on $(-\infty, 0)$ and non-increasing on $[0, \infty)$. In this paper we use the following proximity function.

$$f(x) = \begin{cases} \frac{1}{x^2}, & x < C \\ 0, & \text{otherwise} \end{cases}$$

where the constant C is chosen to be small enough to avoid undesirably long matches during proximity computation.

By definition, given a Chinese query, the operator computes a [0,1] score for every document. Using the positional positional information in the index, the score is made proportional to how proximal the query string characters are in the document under consideration. (Hence the name "proximity" operator.)

It can be noted that this scoring method automatically takes cares of the word boundaries. This follows because if the word boundary were at c_k, then c_k and c_{k+1} will not be adjacent (except under pathological cases) in the document and $Dist_i(c_k, c_{k+1})$ would be zero by definition. Hence the operator would score every document as though the query were properly segmented. An important consequence of this observation is that word segmentation is *not* needed for our method to work. We believe that this method is more natural for Chinese IR than the conventional techniques used by majority of the TREC systems.

We next describe our experiments for TREC 6.

3 Our Experiments

For our first run, we used phrases from the description field and ranked the documents using the proximity based scoring method described above.

Results:

```
Total number of documents over all queries
     Retrieved:    26000
      Relevant:     2958
       Rel_ret:     2215
Interpolated Recall - Precision Averages:
    at 0.00       0.8100
    at 0.10       0.6137
    at 0.20       0.5222
    at 0.30       0.4842
    at 0.40       0.3930
    at 0.50       0.3446
    at 0.60       0.3047
    at 0.70       0.2362
    at 0.80       0.1777
    at 0.90       0.1148
    at 1.00       0.0252
Average precision (non-interpolated) over all rel docs
                  0.3427

Precision:
  At    5 docs:   0.6385
  At   10 docs:   0.6038
  At   15 docs:   0.5821
  At   20 docs:   0.5692
  At   30 docs:   0.5051
  At  100 docs:   0.3715
  At  200 docs:   0.2671
  At  500 docs:   0.1501
  At 1000 docs:   0.0852
R-Precision (precision after R (= num_rel for a query) docs retrieved):
       Exact:     0.3881
```

This is our baseline experiment. (In comparison with our TREC-5 automatic retrieval method[NL96], the new algorithm is observed to show a over 15% improvement in average precision.) This run is called itich3.

For the next experiment, we extracted the top 20 ranked documents for each query from the above and built a 3-gram dictionary from these documents. Rare and common 3-grams were filtered based on term occurrence estimates while building the dictionary. We ranked the entries in the dictionary according to the measure

$$\frac{\text{number of documents containing the phrase}}{\log(\text{number of documents containing the phrase in the whole collection})}$$

and picked top 10 phrases from this dictionary. These phrases are used to expand the query. We assigned a weight of 0.5 for the expanded phrases while the phrases in the original query carried full weight. This run is called itich1.

Results:

```
Total number of documents over all queries
      Retrieved:    26000
       Relevant:     2958
        Rel_ret:     2447
Interpolated Recall - Precision Averages:
    at 0.00        0.8910
    at 0.10        0.7352
    at 0.20        0.6477
    at 0.30        0.6092
    at 0.40        0.5523
    at 0.50        0.4936
    at 0.60        0.4323
    at 0.70        0.3437
    at 0.80        0.2551
    at 0.90        0.1688
    at 1.00        0.0423
Average precision (non-interpolated) over all rel docs
                   0.4541

Precision:
  At    5 docs:    0.7385
  At   10 docs:    0.7192
  At   15 docs:    0.7026
  At   20 docs:    0.6577
  At   30 docs:    0.6115
  At  100 docs:    0.4615
  At  200 docs:    0.3158
  At  500 docs:    0.1644
  At 1000 docs:    0.0941
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:         0.4755
```

The results show that query expansion improves the average precision by over 32%. Appreciable improvement in recall is also observed.

For our last experiment, we expanded phrases as above but filtered out duplicate and common phrases (like dates). This run was called itich2.

Results

```
Total number of documents over all queries
     Retrieved:    26000
      Relevant:     2958
       Rel_ret:     2349
Interpolated Recall - Precision Averages:
    at 0.00       0.8698
    at 0.10       0.7248
    at 0.20       0.6206
    at 0.30       0.5649
    at 0.40       0.4757
    at 0.50       0.4264
    at 0.60       0.3709
    at 0.70       0.3005
    at 0.80       0.2110
    at 0.90       0.1321
    at 1.00       0.0155
Average precision (non-interpolated) over all rel docs
                  0.4145

Precision:
  At    5 docs:   0.7538
  At   10 docs:   0.7192
  At   15 docs:   0.6872
  At   20 docs:   0.6500
  At   30 docs:   0.5885
  At  100 docs:   0.4288
  At  200 docs:   0.2973
  At  500 docs:   0.1590
  At 1000 docs:   0.0903
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.4452
```

The above run, though better than itich1, is actually worse compared to itich1.

An analysis of the performance of three submitted runs is shown below:

itich1 is our best run followed by itich2. We still have no convincing explanation as to why itich2 is poorer than itich1, but we believe it could be due to term weighting. We found that the conventional *tf.idf* weighting coupled with the proximity operator hurts performance. (With just idf weight, however, the performance improves but not significantly on average.) Hence our offical runs included only un-weighted retrieval results. The effect can be seen from Table 3.1. While the Recall@1000 was at or above median for 14 queries for itich1, the avg. precision was above median on only 6 queries. We feel a novel weighting

Run	Average Precision	At or above median on		
		Avg Precision	Recall@100	Recall@1000
itich1	0.4541	6 queries	11 queries	14 queries
itich2	0.4145	4 queries	7 queries	10 queries
itich3	0.3427	4 queries	6 queries	8 queries

Table 3.1: Performance of submitted runs

scheme needs to be devised to work with the proximity operator. We plan to investigate this as part of our future work. Our future investigations will also include more sophisticated proximity operators.

References

[ACC+96] James Allan, Jamie Callan, Bruce Croft, Lisa Ballesteros, John Broglio, Jinxi Xu, and Hongmin Shu. INQUERY at TREC-5. In *Text REtrieval Conference(TREC-5)*. NIST, Gaithersburg, Maryland, 1996.

[BGH+96] M.M. Beaulieu, M. Gatford, Xiangxi Huang, S.E. Robertson, S. Walker, and P. Williams. Okapi at TREC-5. In *Text REtrieval Conference(TREC-5)*. NIST, Gaithersburg, Maryland, 1996.

[BSM96] Chris Buckley, Amit Singhal, and Mandar Mitra. Using query zoning and correlation with SMART: TREC 5. In *Text REtrieval Conference(TREC-5)*. NIST, Gaithersburg, Maryland, 1996.

[GCH+96] Fredric C. Gey, Aitao Chen, Jianzhang He, Liangjie Xu, and Jason Meggs. Term importance, boolean conjunct training, negative terms and foreign language retrieval. In *Text REtrieval Conference(TREC-5)*. NIST, Gaithersburg, Maryland, 1996.

[KG96] K.L. Kwok and L. Grunfeld. TREC-5 English and Chinese retrieval experiments using PIRCS. In *Text REtrieval Conference(TREC-5)*. NIST, Gaithersburg, Maryland, 1996.

[LA96] Joon Ho Lee and Jeong Soo Ahn. Using n-grams for Korean text retrieval. In *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 216–224. ACM, New York, 1996.

[Lin72] Y.T. Lin. *Chinese English Dictionary of Modern Usage.* Chinese University of Hong Kong Press, Hong Kong, 1972.

[NL96] Chong-Wah Ngo and Kok F. Lai. Experiments on Routing, Filtering and Chinese text retrieval in TREC-5. In *Text REtrieval Conference(TREC-5).* NIST, Gaithersburg, Maryland, 1996.

LEXIS-NEXIS
9443 Springboro Pike
Miamisburg, OH 45342
Phone 937.865.6800
Fax 937.865.1655

LEXIS-NEXIS

Query Processing in TREC6

Allan Lu, Ed Meier, Ashwin Rao, David Miller, and Dan Pliske

Allan.Lu, Edwin.Meier, Ashwin.Rao, David.Miller and Daniel.Pliske@Lexis-Nexis.com

October 29, 1997

Introduction

Our ad hoc runs in TREC6 focus on query processing, or to be precise, on short natural language query processing. We investigated the three problems: 1) determining key concept(s) in a short NL query, 2) selecting the synonyms and the related terms for the key concept from WordNet, and 3) identifying useful phrase(s) in the query.
The two ranking algorithms used are derived from Cornell's Lnu.ltu (coded as Panther) and City's BM25 (Ocelot), as we reported in our TREC5 paper [1]. We also used the LEXIS online statistical thesaurus (REL, for RELated terms) for query expansion. After query processing, we employed two relevance feedback strategies, one derived from Rocchio, and another developed internally (Picky). Finally, we investigated the use of a data fusion technique. EUREKA (End User Research Enquiry & Knowledge Acquisition), our research system was used to carry out experiments. EUREKA consists of a rich set of UNIX tools for indexing and ranking.

The three ad hoc runs are LNaShort, LNmShort and LNaVryShort, with "LN" designating Lexis-Nexis, "a" automatic query processing, "m" manual query processing, "Short" the description field in the topics, and "VryShort" the title field. We will cover each run in detail after describing our query process.

Query Processing

Improvement to retrieval effectiveness comes primarily from the two sources: improved ranking techniques (including fusion techniques) [2-4] and improved query enhancement techniques [5-7]. Our focus in TREC6 is on the development of improved of query enhancement techniques.

As we outlined in the introduction, the most critical issue in any query enhancement is the identification of the key concept terms in queries. In the case of some ranking algorithms simply increasing the frequency counts of these key terms will improve the algorithm's effectiveness [7-8]. In the case of other algorithms, the benefit from this sort of frequency manipulation is insignificant (but not harmful). Our approach to identifying key concept terms is simple. Our approach involves using part-of-speech markers to locate nouns and proper names in queries which are taken to be the key terms. This approach is computationally inexpensive and is attractive to us.

Once the key terms are identified, the query expansion takes place. Our expansion technique in TREC6 is query context-sensitive. In our technique terms are obtained from WordNet to perform expansion [9], and are selected based on their statistical correlation with key query terms. We hoped that this query context-sensitive approach would eliminate the problems encountered in earlier studies [10]. In our approach, phrases are viewed as just another type of query term, similar to nouns and proper names.

Automatic Run Using the Description Field

Panther, a derivative of Cornell's Lnu.ltu, was the primary algorithm used in the creation of the LNaShort entry. The main aim in this series of experiments was to assess the effect of query enhancements on effectiveness. We used TREC-4 data as test corpus for these experiments. We experimented with various query enhancement techniques described in the preceding section, and used the results obtained therein to finalize the final data processing steps for TREC-6 data.

After we received the TREC-6 query relevance file, we retraced our steps to see how the various steps affected our performance. In this section, we will elaborate on the various steps taken, and their impact on effectiveness. Most of the discussion below refers to Table 1.

Run	*11_Pnt_Ave*	*Exact_Prec*	*Rel_Ret*	*Top_ 5*	*Top_10*	*Top_20*
Baseline	0.1640	0.2039	1963	0.4040	0.3460	0.2860
+Nouns	0.1642	0.1987	1950	0.4120	0.3540	0.2830
+Phrases	0.1748	0.2132	2033	0.4000	0.3580	0.3010
+Synonyms	0.1737	0.2058	2008	0.4600	0.3780	0.3130
Passage	0.1783	0.2076	1938	0.4040	0.3380	0.2770
OCELOT	0.2046	0.2356	2152	0.4520	0.3920	0.3250
REL	0.1922	0.2312	2243	0.4440	0.3780	0.3190
Fusion-1	0.1980	0.2350	2244	0.4640	0.4120	0.3270
Fusion-1(a)	0.1972	0.2408	2248	0.4560	0.4020	0.3130
Rocchio	0.1869	0.2145	2149	0.4360	0.3780	0.3220
Rocchio(a)	0.1836	0.2120	2120	0.4040	0.3700	0.3100
Fusion-2	0.2022	0.2420	2347	0.4760	0.4040	0.3230
Fusion-2(a)	0.1994	0.2432	2336	0.4720	0.3940	0.3210

Table 1.

First we created a benchmark by ranking the TREC-6 queries without any enhancement. These results are labeled 'Baseline' in Table 1 above. While experimenting with TREC-4 data, we noticed that the presence of particular nouns in queries helped us quickly identify relevant text for further processing. If we highlighted nouns by incrementing their frequencies, we found we could improve effectiveness by approximately 22%. We therefore tried the same technique on TREC-6 data, and surprisingly got quite disappointing results (+Nouns entry in Table 1). This may be attributed to the larger size of TREC-6 queries.

Our next step was to capture the phrases from within the query text. During the TREC-4 experiments, this step seemed to have the most favorable effect out of all the pre-processing steps. We got only 4% increase when we applied this step to TREC-6 data, but as the numbers show, it seemed to be the only pre-processing step that had a positive effect on the number of relevant documents retrieved.

Our final pre-processing step was to add in synonyms of query terms. In the TREC-4 experiments, this step contributed around 20% towards improving the retrieved documents. We were quite disappointed to see that it degraded the performance of the ranking algorithm by a small margin (negligible though it may be).

At this point, we combined the ranking list obtained from three different methods. The other two methods were both based on the OCELOT algorithm. For the results (in Table 1) labeled OCELOT, data was processed through the same stages as described above. The second method used the OCELOT ranking algorithm against the base query along with related terms obtained from REL, the LEXIS-NEXIS online statistical thesaurus. The results of the algorithm is labeled as REL in Table 1. Since different retrieval engines are known to pick up different documents for the same query, we found that the results of the data fusion [11] were quite impressive (Fusion-1 in Table 1).

To perform relevance feedback, we took the top 20 terms from the top 20 documents within Fusion-1, and performed a Rocchio Feedback operation on it to get the results reported as 'Rocchio' in Table 1. Some performance deterioration is observable here, indicating that the relevance feedback did not improve every query. It has been our observation that some queries, due to their higher level of ambiguity or simply due to lack of relevant documents in the database, become lost

in the document space after a blind feedback operation such as traditional Rocchio.

To play it safe and as a final step, we used the Rocchio ranked results and fused them again with the OCELOT and REL results to obtain Fusion-2. The second fusion result indicates that the Rocchio query modification techniques helped by adding more relevant documents to a subset of the queries, hence it improved the final score across the board.

We had initially planned to combine the results of a passage ranking into these two fusion steps, but we could not do so because the passage run did not complete in time. After the TREC deadline, we ran the Passage run to obtain the scores labeled as Passage in Table 1. Precision for the passage run seems to be better than the normal Panther run. But the best part of having the Passage run was that it marginally improved the results of the data fusion. Fusion-1(a) includes the passage run along with the OCELOT, REL and the pre-Fusion-1 Panther run. The score of the Rocchio Feedback operation on the Fusion-1(a) is labeled as Rocchio (a), and the final data fusion operation resulted in Fusion-2(a).

The results above seem to indicate that there were some critical differences between TREC-4 data and TREC-6 data that caused the algorithms to behave in a divergent manner. Panther had an advantage over OCELOT with TREC-4 data because of its superior data length normalization, which did not seem to be a concern in TREC-6. Our next step would be to see if Panther could be modified to improve the ranking on TREC-6 data. We also need to find out why the pre-processing steps deteriorated performance so badly.

Manual Run Using the Description Field

We first ran an automatic short query ranking with Ocelot and then manually scanned the top 20 documents returned. We thoroughly read those documents that seemed relevant, looking for terms that we thought would help the query find more relevant documents. Once we had added the additional terms to the queries we ranked them using Ocelot again to obtain our final ranking list.

Before adding terms to the queries some were edited to remove negating terms. Anything in phrases beginning with “Not”, “Other than”, “Outside the” and the like were deleted so as to make the focus of the

query more straightforward. In addition, we doubled the frequency of the original query words so as to keep the focus on the query subject. We were worried that the manually added words would change the focus of the query to a subject which was similar but not directly relevant.

The table below lists the results of the Manual query run. The Baseline is the query with an extra occurrence count added to capitalized nouns. These were considered proper nouns. By adding the manually selected terms to the Baseline queries we created the +Terms query. Also, the frequency counts of the original words of the query were doubled. The +Nouns query was created by incrementing all words that were determined to be nouns by WordNet. The +Increment was created by incrementing the count of all the words in the query so far. This was done in preparation for the next step. The +Phrases query was created by adding phrases and third person country references.

	11-Pnt_Ave	*Exact_Prec*	*Rel_Ret*	*Top 5*	*Top 10*	*Top 20*
Baseline	0.1859	0.2194	2038	0.3760	0.3400	0.2900
+Terms	0.2883	0.3180	2828	0.6040	0.5020	0.4120
+Nouns	0.2913	0.3133	2822	0.6160	0.5140	0.4180
+Increment	0.2852	0.3014	2781	0.6160	0.5060	0.4140
+Phrases	0.2850	0.2990	2861	0.6160	0.5040	0.4180

TABLE 2: Ocelot Manual Runs

From the table above we can see that the addition of the manually selected terms gave an across the board improvement. Every score and count increased significantly. The additions of noun emphasis, +Nouns, had mixed results. The 11-point average increased. As were the Top n scores. The number of relevant documents returned however, was reduced. As was the Exact Precision score. Incrementing all the counts, +Increment, had disappointing results. All scores were decreased. The addition of phrases and third person country references didn't have a uniformly positive affect either. The number of relevant documents returned hit its highest value. All other scores, though, were equal or worse.

Ocelot Automatic Run

While the Ocelot Automatic Run was not submitted to the TREC 6 competition, they are supplied here for informational purposes. Once again, the Baseline query is the query with an extra occurrence count added to capitalized nouns. These were considered proper nouns. The +Nouns query was created by incrementing all words that were determined to be nouns by WordNet. The +Increment was created by incrementing the count of all the words in the query up to that processing step. This was done in preparation for the next 2 steps. The +Phrases query was created by adding phrases and third person country references. The +Synonyms query was created by adding synonyms determined by WordNet that scored highest in the co-occurrence analysis.

	11_Pnt_Ave	*Exact_Prec*	*Rel_Ret*	*Top_5*	*Top_10*	*Top_20*
Baseline	0.1859	0.2194	2038	0.3760	0.3400	0.2900
+Nouns	0.1867	0.2253	2008	0.4060	0.3640	0.2990
+Increment	0.1932	0.2256	2045	0.4120	0.3600	0.3060
+Phrases	0.2006	0.2308	2128	0.4240	0.3720	0.3230
+Synonyms	0.2046	0. 2356	2152	0.4520	0.3920	0.3250

Exact

TABLE 3: Ocelot Automatic Runs

The table above shows that the addition of nouns to the baseline, +Nouns, was helpful in elevating the relevant documents toward the top of the ranking but decreased the number of relevant documents returned. Incrementing the counts had mostly positive results. The only statistic to drop was the Top 10 and that dropped only marginally. The addition of phrases and third-person country references gave markedly improved scores. The addition of synonyms that scored highly in co-occurrence analysis had a mostly positive impact. The only score to drop was the Exact Precision.

Automatic Run Using the Title Field

The process used here was the simplest among the three submissions. The query process consisted of simply submitting the titles to REL and capturing the suggested terms from REL. REL provides a list of up to 26 words and phrases that are related to one or more title words or phrases. REL uses a statistical thesaurus containing millions of relationships. Words or phrases are considered

related when they co-occur in the same document and are considered important to that document. The final queries for ranking consist of the original title and the suggested terms. The ranking algorithm used in this run is Ocelot with a query term dependency parameter and the relevance feedback technique used is Picky.

Table 4 summarizes the effectiveness of this approach. The base run is the one that used the titles without any further enhancement. The second run employed the word stemming technique, that is, a stemmed item has to be a noun. Some improvement is observable. The third run added the terms related to the original title word or words (no phrase detection was involved in processing the titles) according to REL, and ranked the documents using Ocelot. There is a significant improvement in effectiveness. The fourth run performed relevance feedback using the top 20 documents in this ranking list. Some deterioration is observable, reflecting a diminished return from relevance feedback at this stage (REL expansion may have done the job already). The fourth run, unfortunately, is the run that we submitted to NIST.

The fifth run that takes into account the query word dependency or co-occurrence within a document was planned but never completed due to a software bug discovered in the week before August 15th. A document with a pair of the original title words (not the expanded term(s) from REL) receives a bonus similarity value that is derived from the weights of the two words as well as their physical distance in the title. We believe this approach is less "harsh" than Boolean filtering and provides a better fit into a given ranking algorithm. Some improvement is noticeable. The improvement may be increased if we use different parameter settings. As we observed before, the improvement usually occurs at the top of a rank list, which is a desirable characteristic.

Run	*11_pnt_Ave*	*Exact_Prec*	*Rel_Ret*	*Top_5*	*Top_10*	*Top_20*
Base	0.1524	0.1859	1485	0.3120	0.2920	0.2550
+WdStemming	0.1531	0.1898	1500	0.3360	0.2920	0.2570
+REL	0.2310	0.2656	2375	0.4280	0.3960	0.3420
+Picky	0.2283	0.2648	2396	0.4200	0.3840	0.3310
+TmDpndnt	0.2366	0.2713	2427	0.4480	0.4040	0.3380

TABLE 4: Automatic Title Runs

Summary

The benefit from our query processing is moderate in TREC-6. The process has yet to be fully developed. Several devices such as LEXIS-NEXIS phrase list, LEXIS- NEXIS noisy word list and efficient co-occurrence analysis are still to be placed into the process. Without these devices our processed queries still contain noisy terms or misleading terms that may take the ranking into a different relevance dimension. Little things do add up in our query processing and we need to perfect each of them.

Our manual ad hoc run, which is really an extension to our auto run for simulating a searcher's interaction with an IR system after receiving his initial ranking results, showed very consistent performance across the 50 queries. We will compare our records to those from the interactive group in the conference.

Finally, REL, the LEXIS-NEXIS statistical thesaurus, helped significantly in processing some title-only queries and really destroyed a few others. Regarding phrasing and query dependency constraints, we could do better. We might also be better off with a similar thesaurus compiled using the TREC-6 data sets because of data compatibility.

Note that we did not have any run using the "long" queries. We attempted to do it after the conference in order to make our results more comparable. Unfortunately we could not reassemble the TREC team because everyone was assigned to other development projects.

References

[1] Lu, X. A, Ayoub, M. and Dong, J. "Ad Hoc Experiments Using EUREKA," TREC5 Notebook, 1996.

[2] Singhal, A., Salton, G., Mitra, M. and Buckley, C. "Document length normalization," Information Processing & Management, Vol.32, pp.619-33, 1996.

[3] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M. and Gatford, M. "Okapi at TREC3," The Third Text Retrieval Conference, NIST Special Publication 500-225, pp.109-26, edited by D. Harman, 1995.

[4] Clarke, C. L. A., Cormack, G. V. and Tudhope, E. A. "Relevance ranking for one to three term queries," RIAO'97 Conference Proceedings, pp.388-400, Montreal, Quebec, Canada, 1997.

[5] Lu, X. A., and Keefer, R. B. "Query expansion/reduction and its impact on retrieval effectiveness," The Third Text Retrieval Conference, NIST Special Publication 500-225, pp.231-39, edited by D. Harman, 1995.

[6] Hearst, M. A. "Improving full-text precision on short queries using simple constraints," Proceedings of 5th Annual Symposium on Document Analysis and Information Retrieval, pp.217-32, Las Vegas, Nevada, 1996.

[7] Kwok, K. L. "A new method of weighting query terms for ad-hoc retrieval," In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.187-96, Zurich, Switzerland, 1996.

[8] Allan, J., Callan, J., Croft, B., Ballesteros, L., Broglio, J., Xu, J., and Shu, H. "INQUERY at TREC-5," TREC5 Note Book, 1996.

[9] Miller, G. Special Issue, WordNet: An online lexical database, International Journal of Lexicography, Vol.3 (4), 1990.

[10] Voorhees, E. M. "On expanding query vectors with lexically related words," The Second Text Retrieval Conference, NIST Special Publication 500-215, pp.223-31, 1994.

[11] Lee, J. H. "Analyses of multiple evidence combination," Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.267-77, Philadelphia, PA, 1997.

Query Term Expansion based on Paragraphs of the Relevant Documents

Kai Ishikawa Kenji Satoh Akitoshi Okumura

C&C Media Research Labs.

NEC Corp.

Abstract

Recently, we studied the method of extracting terms that co-occurred with initial query terms in relevant paragraphs as query term expansion method. In our methods, paragraphs in the relevant documents are lanked by using initial query to extract terms from the upper ranked paragraphs with using term co-occurrence.

Our method eases the difficulty by ranking paragraphs with the initial query. Without using term co-occurrence in paragraphs, we could achieve the highly accurate treatment of term co-occurrence by small calculation.

The results of our system for TREC-6 routing test data, obtained by using the expanded queries generated by our query term generation method are compared with the results obtained by initial queries.

1 Introduction

In the field of information retrieval, methods of generating query with extracting query terms from topics(queries written in natural language) and calculating weight from term frequencies [1] are established.

Whereas in case a topic has few clear keywords or it is written in short description, these methods have a problem in composing accurate query with using the terms extracted only from the topic.

To overcome this problem, query term expansion methods are known to be effective, which are query term generation methods that acuire expansion terms from thesaurus or from co-occurrent terms with initial query terms in the relevant documents.

Recently, we studied the method of extracting terms that co-occurred with initial query terms in relevant paragraphs as query term expansion method.

In our methods, paragraphs in the relevant documents are lanked by using initial query to extract terms from the upper ranked paragraphs with using term co-occurrence.

Compared to the methods using term co-occurrence in documents, the methods using term co-occurrence in paragraphs have both advantage and disadvantage: highly accurate treatment of term co-occurrence and huge amount of calculation for each combination of terms in paragraphs.

Our method eases the difficulty by ranking paragraphs with the initial query. Without using term co-occurrence in paragraphs, we could achieve the highly accurate treatment of term co-occurrence by small calculation.

In this paper, we report on our information retrieval system for English documents with using the data of TREC-6 [2] routing task.

First, we explain the query generation methods applied in our system, basic(initial) query generation method by using relevance weighting model [1] and the expanded query generation method by using query term expansion method.

Next, we explain the composisition of modules and process flow of our system.

Finally, we present our results and conclusion of the system using TREC-6 data.

2 Initial Query Generation

Initial queries are generated by extracting query terms from each topic and calculating weight of each term. Here, proper noun phrases are used as query terms, which are extracted from each topic by using dictionary of parsing system.

Term weight is calculated from term frequencies in training data. In the formula, the weight w_i is used for each query term i as shown below [1];

$$w_i = \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)}, \tag{2.1}$$

where

N : the number of documents

n_i : the number of documents containing term i

R : the number of relevant documents

r_i : the number of relevant documents containing term i

Summation of weights w_i for the terms appeared in the document is calculated as the score, which is used to determine the ranking of relevant documents.

3 Expanded Queries Generation

Expanded queries are generated by extraction of expanded query terms and calculation of weight for each term as in the initial query generation process.

First, query terms are extracted from the upper ranked paragraphs of all the documents in training data. Ranking of paragraphs is determined according to the scores that are calculated by using initial queries for each paragraph. Here, proper noun phrases are extracted as expanded terms from the paragraphs, in the same way as in the initial query generation process.

Next, the extracted terms are selected as the extracted query terms which have strong co-occurred with the initial query terms in the upper ranked paragraphs. In this selection, the following mutual information is used to enumerate the strength of the co-occurrence. The mutual information $I(t^I, t^E)$ of the extracted term t^E with the initial query term t^I in the N_B paragraphs is;

$$I(t^I, t^E) = \log \frac{\frac{1}{N_B}\sum_{i=1}^{N_B} T_i(t^I)T_i(t^E)}{(\frac{1}{N_B}\sum_{i=1}^{N_B} T_i(t^I))(\frac{1}{N_B}\sum_{i=1}^{N_B} T_i(t^E))}, \tag{3.2}$$

where $N_B(=15)$ is the number of paragraph, and the function $T_i(t)$ is restricted to take value 1 or 0, according to the term t appears in the paragraph i or not.

After the calculation of the mutual information $I(t^I, t^E)$ of the extracted term t^E with all the initial query terms $\{t^I\}$, the extracted term t^E is selected as the expanded query term if it satisfies the condition $max_{\{t^I\}} I(t^I, t^E) > \gamma, here(\gamma = 2.0)$ is used.

Finally, the expanded query terms are weighted according to the weight value of the initial query terms which has the strongest co-occurrence with them. This weighting is based on the idea that an expanded query term which has strong co-occurrence with an initial query term is supposed to be the relational word(phrase) of the initial query term.

For the calculation of the weight of the expanded query term t^E, the weight value w_{t^I} of the initial query term t^I, which has the strongest co-occurrence with them, is used, i.e.

$$w_{t^E} = \beta w_{t^I}; t^I = argmax_{\{t^I\}} I(t^I, t^E), \tag{3.3}$$

where the parameter $\beta(\beta = 0.15)$ is introduced to control the overall weighting of the expanded query terms.

The values of the parameters $\beta and \gamma$ are determined to optimize the precision of the system, based on the rough estimation using training data.

4 Routing System

4.1 System Composition

Our system for routing task is composed of query generation system and ranking system. Fig.1 shows modules of query generation system.

Here, Training Data, Topic, and Relevance Judgment are initial data of the system.

First, Index File Generation module generates Index File for query term retrieval from training data.

Then, Query Term Extraction module and Weight Calculation module generate Initial Query as their output from Topic and Index File.

Next, Relevant Document Extraction module, Query Term Expansion module and Weight Calculation module generate Expanded Query as their output from Initial Query and Relevant Documents in Training Data.

Finally, Routing Query is obtained by adding Initial Query and Expanded Query.

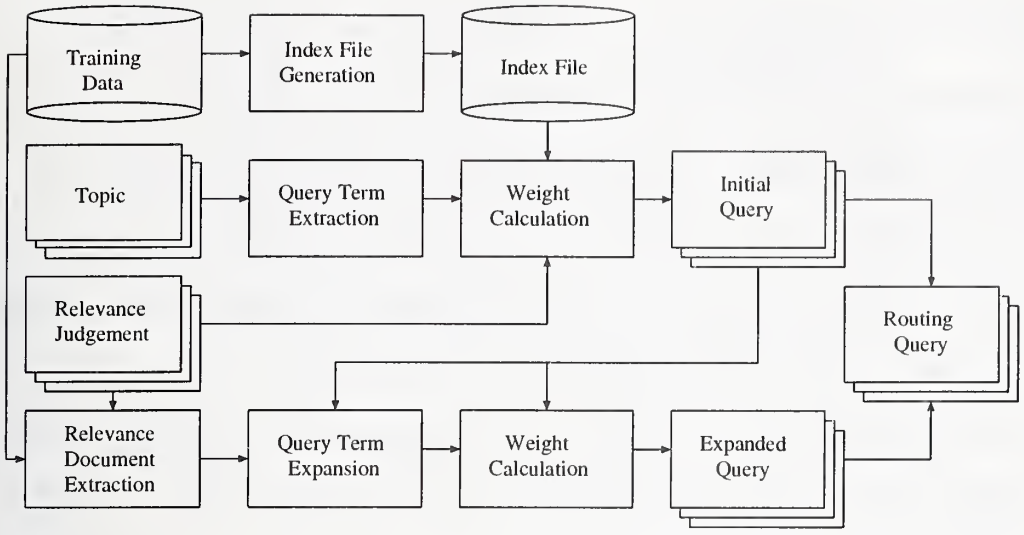


Fig.1 Qery Generation System

Fig.2 shows modules of the ranking generation system.

Here, test data are the initial data of the system. Index File Generation module generates index files for query term retrieval from test data.

Finally, Score Calculation module generates rankings of documents as output from routing queries and index files.

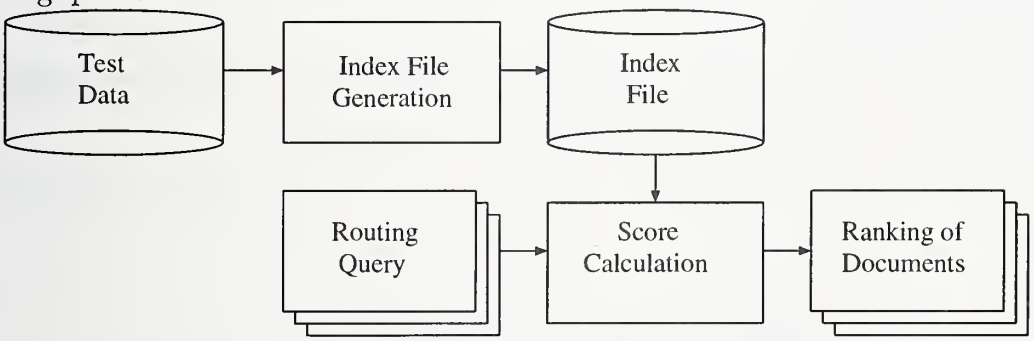


Fig.2 ranking generation system

4.2 Process Flow

Here, the process flow of the routing task is shown.

1. Initial query generation

   (a) Extraction of noun phrases from each topics as initial query terms

   (b) Weighting of query terms by formula (2.1)

2. Expanded query generation

   (a) Extraction of paragraphs from relevant documents (to topics)

   (b) Ranking paragraphs by their score calculated from initial query

   (c) Extraction of noun phrases from the upper ranks of paragraphs

   (d) Selection of expanded query terms from noun phrases who co-occurred strongly with initial query terms in the upper ranks of paragraphs

   (e) Weighting of the expanded query terms according to the weights of initial query terms who co-occurred strongly with the expanded query terms .

3. Ranking generation

   (a) Generation of the Routing Query by adding Initial Query and Expanded Query

   (b) Ranking of the documents according to their score calculated by routing query

5 Results

In this section, our results for TREC-6 routing test data, obtained by using two types of queries, the initial queries and the expanded queries, are compared.

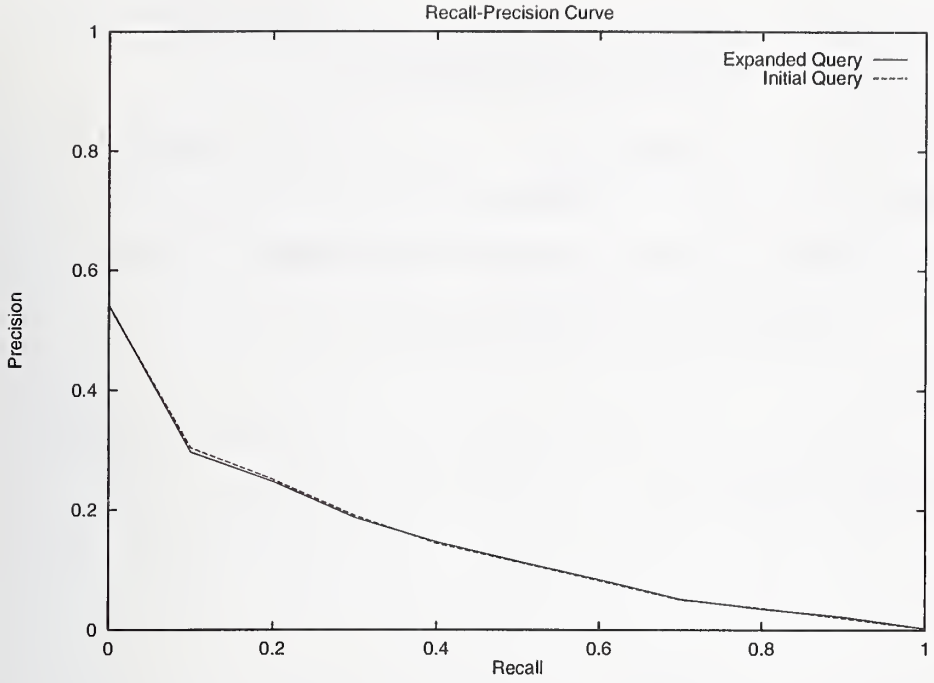
Table 1. shows the values of Average Precision and R-precision calculated from the results obtained by using the initial queries and the expanded queries.

Table 1. Results of our systems

	Average Precision	R-precision
Initial Query	0.1337	0.1815
Expanded Query	0.1328	0.1823

Fig. 3. shows the Recall-Precision curve calculated from the results obtained by using the initial queries and the expanded queries.

Fig. 3. Results of our systems



These values and curves show that there are small difference between the results obtained by using the initial queries and the expanded queries. For the TREC-6 routing test data, effectiveness of the query term expansion cannot be admitted from the above results.

6 Conclusion

We obtained results of our system by using two types of queries, the initial queries and the expanded queries for TREC-6 routing test data. However, effectiveness of the query term expansion for the TREC-6 routing test data cannot be admitted from these results. This results is the opposite of the another results, which we obtained by using training data.

The problem of the query term expansion for the TREC-6 routing test data is caused by the inexistence of expanded terms in test data extracted from training data.

A solution of this problem is to determine appropriate value of paragraphs(N_B) ,because this inexistence is caused by small value of $N_B = 15$used in query term expansion.

By increasing the value N_B, number of terms appeared frequently in irrelevant docu-

ments increase, which decrease the precision of the system. This problem can be solved by considering the strength of co-occurrency in irrelevant documents.

The weighting method for expanded query terms is based on the assumption that a term in relevant paragraphs that co-occurrate strongly with a initial query term is the closely related term of the query term in the topic. This method should be examined by comparing with the relevance weighting method to show its effectiveness.

As our future development, we will examine the above problems of our query term expansion method by comparing with the other query term expansion methods recently applied widely by other TREC participants.

References

[1] Robertson, S.E. and Sparck Jones, K. Relevance weighting of search terms. Journal of the American Society for Information Science,27,pages 129-146,1976.

[2] http://potomac.ncsl.nist.gov/TREC/

[3] Kenji Satoh, Akitoshi Okumura, Kiyoshi YAMABANA. Information Retrieval System for TREC3. The Third Text REtrieval Conference(TREC-3),pages 311-318.

[4] Kenji Satoh, Susumu Akamine and Akitoshi Okumura. Improvements on Query Term Expansion and Ranking Formula. The Fourth Text REtrieval Conference(TREC-4),pages 475-481.

A Comparison of Boolean and Natural Language Searching for the TREC-6 Interactive Task

William Hersh
Bikram Day

Division of Medical Informatics and Outcomes Research
School of Medicine
Oregon Health Sciences University

Abstract

The TREC-6 interactive task used a multi-site experimental protocol, where each participating site compared an "experimental" system with a common "control" system used at all sites. For the Oregon Health Sciences University site, the "experimental" system was a Boolean interface to the MG system, while the control system was, as for all sites, the natural language ZPRISE system. Performance was measured by aspectual recall and precision. OHSU searchers did well overall, achieving the highest overall aspectual precision. These searchers did obtain below-average aspectual recall overall, although they achieved above-average aspectual recall with the control system, indicating that for the TREC-6 interactive task, a natural language searching system was superior to a Boolean one.

Background

A long-standing research issue of interest to information retrieval (IR) researchers at Oregon Health Sciences University (OHSU) is whether end-user searchers achieve better results with Boolean or natural language searching. Previous research in this area is decidedly mixed. The first study to compare Boolean and natural language searching with real searchers was the CIRT study, which found roughly comparable performance between the two when utilized by search intermediaries (Robertson and Thompson 1990). Turtle found, however, that expert searchers using a large legal database obtained better results with natural language searching (Turtle 1994). We have performed several studies of medical end-user searching comparing Boolean and natural language approaches. Whether using recall-precision metrics in bibliographic (Hersh, Buckley et al. 1994) or full-text databases (Hersh and Hickam 1995), or using task-completion studies in bibliographic (Hersh, Pentecost et al. 1996) or full-text databases (Hersh, Elliot et al. 1995), the results have been comparable for both types of systems.

The TREC experiments have been an important landmark for IR experimentation, providing a common large database and set of queries for the entire research community. We therefore chose to use the TREC-6 interactive task to compare systems that featured Boolean and natural language searching. Since the "control" system for the task was ZPRISE, a natural language system, we used a Boolean system for our "experimental" system. The system we chose to use was a Web-based Boolean interface that has been developed for MG, an experimental retrieval system developed at the Royal Melbourne Institute of Technology (RMIT) (Witten, Moffat et al. 1994). MG has a natural language searching interface as well, which was not used for this

experiment. Therefore the results are more applicable to the Boolean searching interface, which we designate in the paper as MG-B, than the MG system as a whole.

Methods

Systems

Both the "experimental" MG-B and "control" ZPRISE systems were compiled and run on a Sun Ultrasparc 140 with 256 megabytes of RAM running the Solaris 2.5.1 operating system. Each system ran as a server, with MG accessed by a CGI form from a Web page and ZPRISE accessed using its Z39.50 client software.

The systems were accessed by Compaq DeskPro 200 MHz Pentium Pro machines running Windows 95. For PRISE, the Xwin32 X-Windows software package (StarNet Inc., Sunnyvale, CA) was used. MG was accessed via Netscape Navigator 3.0 via the Boolean interface available from the MG Web site that we modified slightly to allow logging and to make some minor cosmetic changes. Figure 1 shows the MG-B interface. The interface performs an OR between words on the same line and an AND between lines.

Experimental Design

The details of the overall interactive task are described elsewhere in the proceedings. We describe here the experimental design at OHSU. The searchers were librarians or researchers who work in our IR research laboratory. None of the searchers was familiar with the experimental design coming into the experimental session, and none of them had ever used ZPRISE or MG.

Each searcher was given a series of questionnaires that had been circulated by email from the Rutgers group. A pre-experiment questionnaire asked users about demographic information and past searching experience. A post-search questionnaire was given after each search, while a post-experiment questionnaire was given at the end of the session to obtain feedback about the experiment and searching systems that were used.

Results

Searchers

Table 1 characterizes the searchers based on the information provided on the pre-experiment questionnaire. Even though none of the searchers had used the ZPRISE or MG systems before, all had substantial searching experience. One searcher was actually a librarian, while the other three have done substantial Web searching over the last several years.

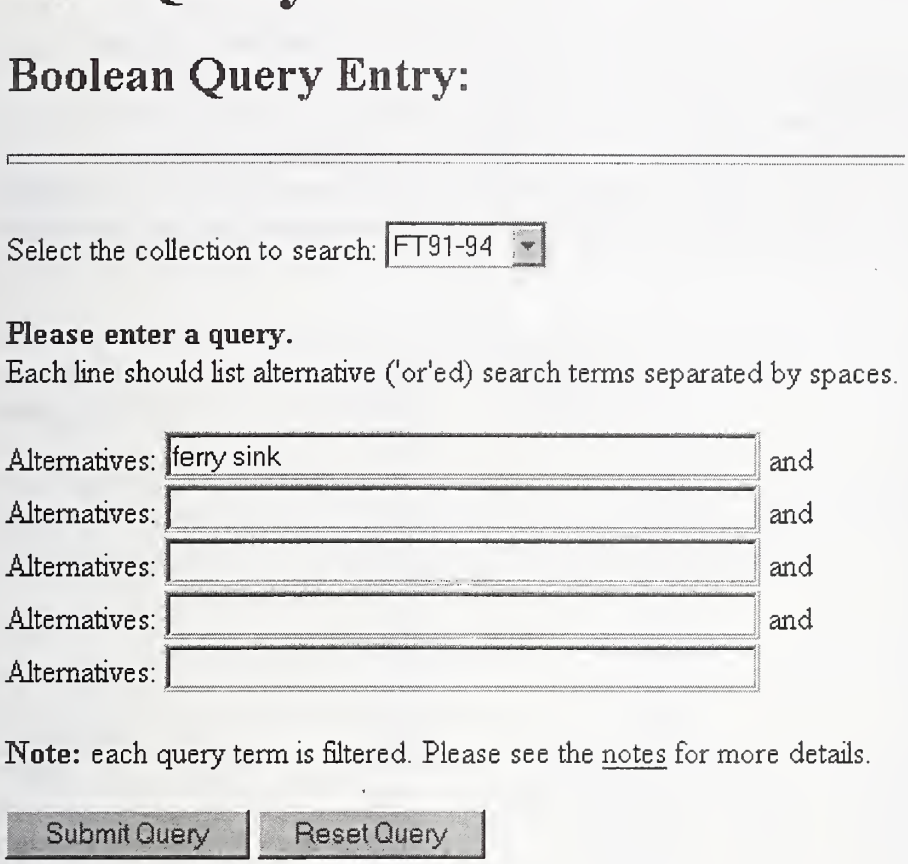


Figure 1 – MG Boolean Web interface.

Searcher	LD	LS	KS	SM
Highest degree	Masters	Masters	Bachelors	Masters
Age	41-50	41-50	21-30	31-40
Gender	M	F	M	F
Years searching	8	15	4	5
Experience searching (1-5)				
Library catalogs	5	5	4	3
CD-ROMs	4	5	2	1
Commercial on-line	2	4	1	3
Web browsers	5	5	5	5
Full-text databases	5	4	1	2
Ranked retrieval systems	5	5	1	4
Rel. Feedback sysems	3	3	5	1
Mouse-based interface	5	5	5	5
Occupation	Programmer	Librarian	Student	Programmer

Table 1 – OHSU searcher characteristics. Experience was rated on a 1 to 5 scale, with 1 indicating no experience and 5 a great deal.

Aspectual recall and precision

Table 2 shows the aspectual recall and precision for all of the groups participating in the task, both combined and broken down by experimental and control systems. The OHSU group obtained the highest aspectual precision of any group (0.85, mean 0.73, range 0.63-0.85). OHSU's overall aspectual recall was on the lower end (0.43, mean 0.45, range 0.32-0.52), although this was mainly due to the poor performance of our experimental system. The OHSU searchers were among the top performers in aspectual recall with the control ZPRISE system (0.49, mean 0.45, range 0.38-0.51).

Precision			
Site	**Both**	**Experimental**	**Control**
BrklyINT	0.80	0.79	0.80
city	0.76	0.71	0.81
IBM	0.63	0.62	0.64
INQ4iai	0.64	0.63	0.64
INQ4iaip	0.76	0.75	0.77
INQ4int	0.67	0.67	NA
NMSU	0.83	0.82	0.83
OHSU	0.85	0.90	0.81
rmit	0.79	0.80	0.78
rutint1	0.67	0.67	NA
rutint2	0.66	0.66	NA
unc6ia	0.67	0.60	0.75
unc6ip	0.78	0.79	0.77
MEAN	**0.73**	**0.72**	**0.76**

Recall			
Site	**Both**	**Experimental**	**Control**
BrklyINT	0.53	0.57	0.49
city	0.39	0.40	0.38
IBM	0.32	0.26	0.38
INQ4iai	0.41	0.36	0.45
INQ4iaip	0.47	0.50	0.44
INQ4int	0.48	0.48	NA
NMSU	0.46	0.47	0.45
OHSU	0.43	0.37	0.49
rmit	0.48	0.47	0.50
rutint1	0.46	0.46	NA
rutint2	0.53	0.53	NA
unc6ia	0.48	0.44	0.51
unc6ip	0.46	0.47	0.46
MEAN	**0.45**	**0.45**	**0.45**

Table 2 – Aspectual precision and recall values for both systems, the experimental system, and the control system for each site.

Table 3 shows the aspectual recall and precision by topic. OHSU searchers tended to follow the average trends.

Table 4 shows the results of individual OHSU searchers. An analysis of variance was done for recall and precision with a three-factor model based on topic, searcher, and system. There were no statistically significant differences in any factor for precision, but for recall there was definite statistical significance between topics ($p < .0001$), near significance between systems ($p = .07$), and no significance between searchers.

Table 5 shows the results of the post-experiment questionnaire. All of the searchers understood the task. They also had a tendency to believe the systems were different. ZPRISE was generally ranked as easier to use than MG-B.

The appendix of this paper contains a detailed narrative of one searcher (KS) for one topic (326i), as required for the interactive site reports. We include this narrative because it highlights some of the issues that arise in interactive searching experiments.

Precision

Topic	All	E	C	OHSU-All	OHSU E	OHSU C
303i	0.71	0.70	0.73	0.85	0.88	0.83
307i	0.79	0.79	0.79	0.85	0.70	1.00
322i	0.45	0.37	0.56	0.85	1.00	0.71
326i	0.92	0.93	0.90	0.92	1.00	0.83
339i	0.79	0.79	0.79	0.80	0.80	0.80
347i	0.74	0.71	0.77	0.83	1.00	0.67

Recall

Topic	All	E	C	OHSU-All	OHSU E	OHSU C
303i	0.89	0.86	0.93	1.00	1.00	1.00
307i	0.30	0.33	0.26	0.29	0.30	0.28
322i	0.11	0.10	0.13	0.19	0.11	0.28
326i	0.44	0.50	0.37	0.39	0.39	0.39
339i	0.72	0.69	0.75	0.55	0.30	0.80
347i	0.21	0.18	0.25	0.15	0.13	0.17

Table 3 – Aspectual precision and recall by query for all sites and OHSU.

Topic	Searcher	System	Recall	Precision
303i	KS	ZP	1.00	1.00
303i	LD	ZP	1.00	0.67
303i	LS	MG-B	1.00	0.75
303i	SM	MG-B	1.00	1.00
307i	KS	MG-B	0.39	0.69
307i	LD	MG-B	0.22	0.71
307i	LS	ZP	0.35	1.00
307i	SM	ZP	0.22	1.00
322i	KS	MG-B	0.11	1.00
322i	LD	MG-B	0.11	1.00
322i	LS	ZP	0.22	0.67
322i	SM	ZP	0.33	0.75
326i	KS	MG-B	0.33	1.00
326i	LD	MG-B	0.44	1.00
326i	LS	ZP	0.44	1.00
326i	SM	ZP	0.33	0.67
339i	KS	ZP	0.70	0.60
339i	LD	ZP	0.90	1.00
339i	LS	MG-B	0.50	0.60
339i	SM	MG-B	0.10	1.00
347i	KS	ZP	0.15	0.50
347i	LD	ZP	0.19	0.83
347i	LS	MG-B	0.12	1.00
347i	SM	MG-B	0.15	1.00

Table 4 – Aspectual recall and precision for OHSU searchers by topic-system pair.

Searcher	LD	LS	KS	SM
Understand task (1-5)	4	5	5	5
Task similar to others (1-5)	3	5	5	1
Systems different (1-5)	4	3	4	5
System comparisons				
Easier to use	MG-B	ZP	ZP	ZP
Easier to learn	MG-B	ZP	ZP	ZP
Liked best	ZP	ZP	MG-B	ZP

Table 5 – Post-experiment questionnaire. Leikert-scale ratings from 1 (not at all) to 5 (complete).

Conclusions

The TREC-6 interactive task was designed to be a multi-site experiment. As with most other sites, there was a small sample size at the individual OHSU site, which makes generalization of the results difficult. However, these experiments did show that OHSU searchers in general did well on the TREC-6 interactive task, and that natural language searching performed better than Boolean searching.

Acknowledgments

The authors would like to thank Paul Over at NIST for all of his assistance in getting ZPRISE to run, managing the experiments, and helping with the collection of data.

References

Hersh, W., C. Buckley, et al. (1994). OHSUMED: an interactive retrieval evaluation and new large test collection for research. *Proceedings of the 17th Annual International ACM Special Interest Group in Information Retrieval*, Dublin, 192-201.

Hersh, W., D. Elliot, et al. (1995). Towards new measures of information retrieval evaluation. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, 164-170.

Hersh, W. and D. Hickam (1995). An evaluation of interactive Boolean and natural language searching with an on-line medical textbook. *Journal of the American Society for Information Science,* 46: 478-489.

Hersh, W., J. Pentecost, et al. (1996). A task-oriented approach to information retrieval evaluation. *Journal of the American Society for Information Science,* 47: 50-56.

Robertson, S. and C. Thompson (1990). Weighted searching: the CIRT experiment. *Informatics 10: Prospects for Intelligent Retrieval*, York, 153-166.

Turtle, H. (1994). Natural language vs. boolean query evaluation: a comparison of retrieval performance. *Proceedings of the 17th Annual International ACM Special Interest Group in Information Retrieval*, 212-220.

Witten, I., A. Moffat, et al. (1994). *Managing Gigabytes - Compressing and Indexing Documents and Images*. New York, Van Nostrand Reinhold.

Appendix – Narrative Description of Topic 326i for One Searcher

The OHSU "experimental" system was a Boolean interface on top of the MG system. We used a simple Web interface that comes with the software distribution. This interface performs an OR between all words on an individual line and an AND between lines. Results are displayed in arbitrary (i.e., non-ranked) order. This document represents our narrative description for one searcher on topic 326i.

Topic

```
<num> Number: 326i
<title> Ferry Sinkings

<desc> Description:
Any report of a ferry sinking where 100 or more people lost
their lives.

<narr> Narrative:
To be relevant, a document must identify a ferry that has
sunk causing the death of 100 or more humans.  It must
identify the ferry by name or place where the sinking
occurred.  Details of the cause of the sinking would be
helpful but are not necessary to be relevant.  A reference
to a ferry sinking without the number of deaths would not be
relevant.

<aspects> Aspects:
Please save at least one RELEVANT document that identifies EACH DIFFERENT
ferry sinking of the sort described above. If one document document discusses
several such sinkings, then you need not save other documents that repeat
those aspects, since your goal is to identify different sinkings of the sort
described above.
```

Searches

The searcher entered 6 searches into the system. Unfortunately, our logging function for MG did not record the Boolean operators and only saved the stemmed version of terms entered (i.e., not the original query text). However, it is apparent from the results below that the searcher used at least one OR for the first three searches and ANDs for the last three searchers.

Time	Input	Docs Viewed	Docs Seen
1:11:17 PM	fer sink	666	0
1:11:53 PM	fer sink ship 100 dead	234	0
1:12:41 PM	fer sink ship	1148	1
1:14:25 PM	fer sink	53	9
1:26:30 PM	fer traged	22	2
1:28:56 PM	fer sunk	14	1

Documents Viewed

A total of 2,137 documents were viewed, 1,468 of which were unique. The overwhelming majority of these were retrieved by the first three searches.

Documents Seen

The searcher only chose to see one document from the results of the first three searches, and was likely overwhelmed by the massive output. After the fourth search, the searcher began choosing documents to see.

Time	Document
1:13:42 PM	FT911-3535
1:17:02 PM	FT931-8485
1:17:02 PM	FT943-312
1:17:02 PM	FT943-316
1:17:02 PM	FT943-535
1:17:02 PM	FT943-536
1:17:02 PM	FT943-543
1:17:02 PM	FT944-15661
1:17:02 PM	FT944-18722
1:17:02 PM	FT944-18877
1:27:10 PM	FT931-16406
1:27:10 PM	FT944-18719
1:29:28 PM	FT944-18938

Searcher Aspects

The searcher selected six aspects. He actually did not follow the instructions correctly, as the latter 5 aspects represent a single ferry sinking. In fact, he did not explicitly identify NIST aspect 5, the Belgian sinking, but got credit for it because one of his designated documents describes that sinking as well.

Sequence	Document	Text
1	FT931-8485	Crowded Ferry Sinks off Haiti
2	FT943-316	Ferries face calls for safety curbs; Estonia disaster brings
3	FT943-536	Safety Rules (Estonia)
4	FT943-543	Bow Doors Leak Reported (Estonia)
5	FT944-18722	Review of Emergency Procedures (Estonia)
6	FT944-18722	Reactions to the Disaster (Estonia)

NIST Aspects

The table below lists the aspects identified by NIST assessors.

Sequence	Text
1	Zairean ferry accident
2	Neptune ferry sinks
3	Korean ferry sinks west coast of South Korea
4	Moby Prince ferry fire - (2.5 miles at sea)
5	Herald Free Enterprise ferry off Belgian coast
6	Ferry capsizes Port of Mombasa
7	Estonian sinks
8	Bangladesh ferrys sink in Bay of Bengal (Oct 94)
9	Philippine ferry sinking (apparently in the Philippines)

NIST Aspects Coverage

The table below lists the documents and aspects covered as identified by NIST assessors.

Document	Coverage
FT921-924	000000000
FT922-6072	000000000
FT923-4546	000000000
FT923-6558	000000000
FT931-5947	100000000
FT931-8485	010000000
FT934-15680	001000000
FT934-1954	000110000
FT941-9683	000000000
FT942-12305	000001000
FT942-14757	000000000
FT943-178	000010100
FT943-312	000000100
FT943-316	000000100
FT943-3240	000010000
FT943-3945	000000000
FT943-3954	000000000
FT943-535	000000100
FT943-536	000010100
FT943-543	000010100
FT943-569	000000000
FT944-10102	000000100
FT944-10109	000000100
FT944-10853	000000000
FT944-11013	000000100
FT944-11048	000010100
FT944-11367	000000000
FT944-12822	000000000
FT944-15057	000000100
FT944-15143	000000100

FT944-15661	000000010
FT944-1600	000000100
FT944-17957	000000100
FT944-17958	000000000
FT944-18204	000000100
FT944-18217	000000000
FT944-18499	000000100
FT944-18508	000000100
FT944-18722	000000100
FT944-18875	000010100
FT944-18938	000000100
FT944-3393	000000000
FT944-5084	000000100
FT944-5248	000000101
FT944-5773	000000001
FT944-6361	000000000
FT944-6974	000000100

Conventional Recall and Precision

We attempted to apply conventional (i.e., document level, not aspectual) recall and precision calculations to the data. We grouped documents based on the one or more aspects they contained. As is seen below, the searcher identified aspects 2, 5, and 7. He viewed and saw all documents with these aspects. He also viewed and saw the document with aspect 8, but did not designate it as an aspect. He did not view nor see the documents with aspects 1, 3, 4, 6, and 9.

Of the 26 "relevant" documents, he retrieved 23. By standard calculations, his recall would be 88.5%. Calculating his precision is more difficult, due to the large number of documents viewed by the initial Boolean searches. It is unclear whether or not to include those.

Aspect(s)	Relevant	Viewed/Seen
1	1	0
2	1	1
3	1	0
4&5	1	0
5	1	1
5&7	3	3
6	1	0
7	15	15
8	1	1
9	1	0

Rutgers' TREC-6 Interactive Track Experience

N.J. Belkin, J. Perez Carballo, C. Cool[°], S. Lin, S.Y. Park,
S.Y. Rieh, P. Savage, C. Sikora, H. Xie and J. Allan[*]

School of Communication, Information & Library Studies
Rutgers University
4 Huntington Street
New Brunswick, NJ 08901-1071

Abstract

The goal of the Rutgers TREC-6 Interactive Track study was to compare the performance and usability of a system offering positive relevance feedback with one offering positive and negative relevance feedback. Our hypothesis was that the latter system would better support the aspect identification task than the former. Although aspectual recall was higher for the system supporting both kinds of relevance feedback (0.53 vs. 0.46), the difference was not significant, possibly because of the small number of subjects (four in each condition, each doing three searches). Usability results were also equivocal, perhaps due to the complexity of the system. Compared to ZPRISE, the control system without relevance feedback, both relevance feedback systems were rated more difficult to learn to use, but more effective.

1. Introduction

The focus of the Rutgers TREC-6 Interactive Track study was investigating the effectiveness and usability of negative relevance feedback (RF) in interactive information retrieval (IR). This followed from the results of our TREC-4 and TREC-5 studies, in which our subjects expressed a desire to be able to control retrieval in order to suppress documents they did not like. This led us to hypothesize that supporting negative as well as positive relevance judgments in interactive IR would lead to improvements in performance of various IR tasks. These results, and the manner of implementation of negative RF which we think follow from them, are reported in Cool, Belkin & Koenemann (1996).

Briefly, we suggest that there are basically two ways in which negative relevance judgments can be understood in the context of automatic RF. The first, which we call the "classical" model, assumes that terms which appear in the query and positively judged documents, and also in negatively judged documents are "poor" terms from the point of view of IR, since they are bad discriminators. This model therefore reduces the query-term weight of such terms until they reach zero weight, when they are removed from the query. Experiments in non-interactive environments have shown that using negative weights decreases performance. In this model, there is generally no account taken of terms which appear only in negatively judged documents.

[°] Graduate School of Library & Information Studies, Queens College, CUNY

[*] Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts at Amherst.

In contrast to the classic model, we propose an alternative interpretation, which assumes that terms in the original query, and which are added to that query through positive RF are "good" terms whether or not they also appear in negatively judged documents, since they are indicators of what the searcher is looking for. The meaning of the negative judgment in this model is understood to be that the *context* in which the good terms appear is inappropriate to the searcher's problem, or that the topic which they represent is treated only peripherally, or from an inappropriate point of view. Thus, important terms in the negatively judged documents, which do not appear in positively judged documents, are understood as indicators of the inappropriate context, or the main topic, or the inappropriate point of view. This model thus leads us to a quite different way to implement RF, in which query terms which appear in positively judged documents (irrespective of their appearance in negatively judged documents) have their query-term weights increased, and in which the query is expanded by both the important terms in the positively judged documents (with positive weights) and by the important terms in the negatively judged documents which do not appear in the query or the positively judged documents (with negative weights).

The TREC-6 Interactive Track task of identifying the different aspects of a topic offers an especially good environment to investigate the effectiveness of the type of RF our model suggests. Our hypothesis is that once a searcher has identified some aspect of a topic in a particular document, a negative relevance judgment on that document will depress the retrieval status value (RSV) of other documents on the topic which treat that specific aspect, thus promoting documents which treat different aspects of the topic in the output ranking, making it easier to find these new aspects. On the other hand, positive relevance judgments will tend to increase the RSV of other documents which treat the same aspect of the topic, thus demoting documents treating different aspects of the topic in the output ranking, making it more difficult to find new aspects.

Following the results of Koenemann (1996) and Koenemann & Belkin (1996), which suggest that user control of RF leads to enhanced performance and usability, we implemented RF in both of our experimental systems as a term-suggestion device for query expansion, rather than as an automatic query modification device. Thus, the terms which would be added through automatic RF were displayed to the searcher as each relevance judgment was made, for the searcher to choose from for adding to the query (as either a positive or a negative term). The interface and the details of the implementation are described more fully in section 2.

We ran our experiment according to the TREC-6 Interactive Track protocol, with four subjects searching on the control system (ZPRISE) and the positive RF system (ruinq1), and four subjects searching on the control system (ZPRISE) and the positive plus negative RF system (ruinq2). Unfortunately, we goofed and did not log the ZPRISE searches in either of these conditions. Thus, although we can compare subjective judgments of the three systems, we cannot compare performance of either of our experimental systems with the experimental systems of the other participants in the Interactive Track, since they can be strictly compared only through *differences* in performance on the control and experimental system(s) at each site, not the absolute performance on any measure.

2.0 Methods

In this section we describe the research methods we used in conducting our experiment, along with a description of the systems themselves.

2.1 Research Methods

Eight volunteer searchers were recruited to participate in the study, from the population of students in the School of Communication, Information and Library Studies at Rutgers University and from the larger community of information professionals in the New Jersey area. As a condition of the study, none of the participants had taken part in previous TREC studies and none had any prior experience with either RU-INQUERY or ZPRISE. The general demographic characteristics of the searchers and their experiences with IR systems are described below in section 3.1.

Each searcher performed six searches on six topics: three of the searches on a control system (ZPRISE) and three on an experimental system (RU-INQUERY). Searchers were alternately assigned to one of two versions of the RU-INQUERY system. Version 1 (E1) offered positive relevance feedback only; while version 2 (E2) offered both positive and negative relevance feedback. Using ZPRISE as a control system (C), the searchers were randomly assigned to one of the following conditions: E1 and C; C and E1; E2 and C; C and E2. We replicated the conditions twice, using a single ordering of topics.

Before conducting their searches, participants completed a self-administered questionnaire which asked about their demographic characteristics and their searching experiences with a variety of IR systems. They then received a 20-minute interactive tutorial for each of the IR systems, prior to searching on them. Searchers were given 20 minutes to conduct each of their six searches. After each search, subjects answered several questions about their familiarity with the search topic, experiences with the searching task, and their satisfaction with the results. Each search was videotaped, and computer logged. Participants were instructed to "think aloud" about what they were doing, and why, as they searched and these verbal protocols were captured on the videotapes. This process was repeated six times, across the two systems. At the end of this entire session, searchers completed an exit interview which focused on their understanding and use of relevance feedback; their perceptions of the utility of RF for the aspect retrieval task; and their experiences with the IR systems.

2.2. Systems

We used InQuery 3.1p1 as the basis for our experimental systems and the ZPRISE Interactive Track Release as the control system. The two versions of InQuery are: 1) the positive relevance feedback only system (ruinq1); and 2) the positive and negative relevance feedback system (ruinq2). Both of these used the default indexing of InQuery 3.1p1, the Porter stemmer, and the default weighting and matching functions. User query formulation was restricted to unstructured queries, plus the phrase operator (instantiated by enclosing the phrase words within double quotes). RF query expansion (for both positive and negative RF) was implemented using the default InQuery 3.1p1 term ranking formula (tf*rdf), with the number of suggested terms determined by the formula:

$5n + 5$, where n = number of judged documents

to a maximum of 25 suggested terms. The query was parsed as a weighted sum, using the default weighting for RF term addition for positive terms, and adding the negative terms under the InQuery "NOT" operator, with 0.6 weight. Appendix A is a screen dump of the ruinq2 interface; the ruinq1 interface is identical, except that the frames in the lower left and upper right of the interface (those having to do with negative term suggestion, and negative term addition, respectively) are removed, and there are no negative RF buttons.

The functions that are offered by the systems are:

1. Unstructured query input plus phrases in the query formulation window (top center frame);
2. Saving, clearing and loading queries;
3. Display of rank, date and title of ten retrieved documents at a time (center frame);
4. Scrolling the title display ten documents at a time;
5. Saving a document to indicate one or more aspects - unsaving by clicking on saved document button (right hand button on the title line);
6. Marking a document relevant or not relevant to get term suggestions - unmarking by clicking on relevant or nonrelevant document button (two left buttons on the title line). Unmarking removes the document from the RF pool and thus changes the appropriate term suggestion display, but does not affect the selected terms;
7. Display of suggested RF query expansion terms (positive terms displayed in upper leftmost frame; negative terms displayed in lower leftmost frame);
8. User selection of suggested terms to be added to the query by clicking on the desired term (displayed in the top rightmost frame for negative terms, the immediately adjacent frame for positive terms);
9. User deselection of RF terms by clicking on the desired term in the appropriate selected term frame (deselected terms returned to the appropriate term suggestion frame);
10. Clearing all relevance markings (removes all term suggestions, but not term selections);
11. Displaying the full text of a document by double clicking on the title line (displayed in the bottom center frame);
12. Scrolling through the full text of the document;
13. Highlighting query terms in the full text display;
14. Scrolling directly to the next query term in full text display (Show Next Keyword);
15. Showing the best (next best, previous best) passage in the full text display, according to default InQuery 3.1p1 method;
16. Displaying the full text of the next document or the previous document in the retrieved list.

Marking a document saved (unsaved) and relevant or not relevant (or unmarking) is indicated by toggling change in color of the relevant button. Relevant was indicated by green, not relevant by red, and the terms in the term suggestion and selected terms frames were in the same colors.

All three systems ran on a SUN Ultra 140 with 64MB memory and 9GB disk under Solaris 2.5.1, using a 17" color monitor.

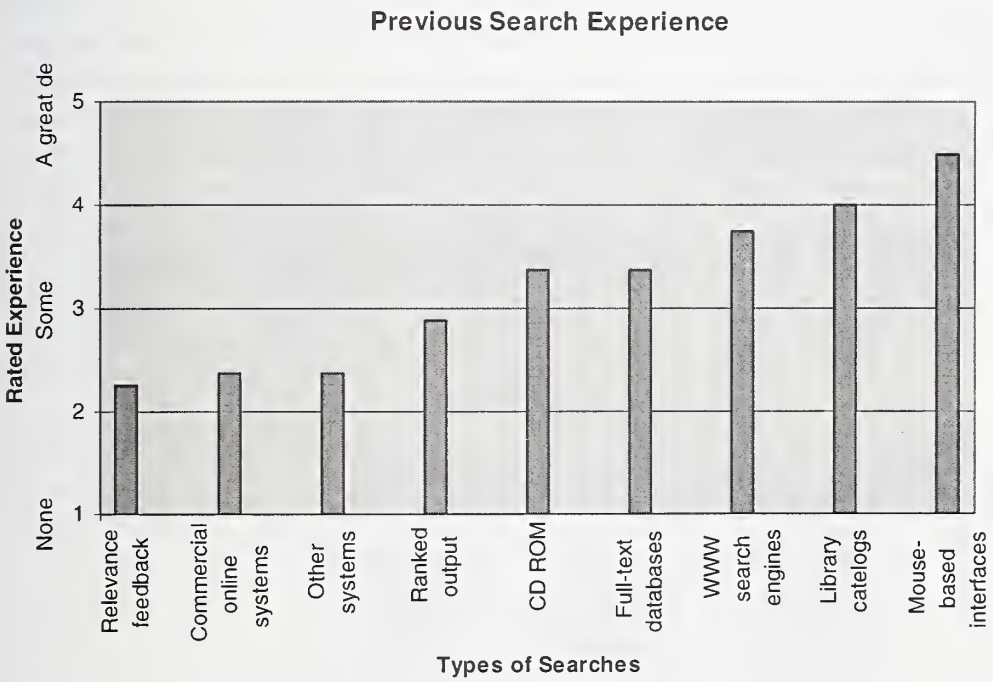
3.0 Results

In the following sections we report on our analyses of the questionnaire and interview data, followed by a discussion of our performance results.

3.1 Characteristics of the Searchers

Our subject group included 6 females and 2 males. The subjects were distributed fairly evenly across age categories with the youngest being under 21 and the oldest between 51 and 60. Five of the eight subjects had, or were pursuing, a graduate degree in library science. The other three subjects indicated no education in library science and had, or were pursuing, Bachelor degrees in other fields. As mentioned above, none of the subjects reported participating in any previous TREC experiments or having any previous experience with the ZPRISE or RU-INQUERY information retrieval systems. The median number of years reported for overall experience doing online searching was three and a half ($\underline{M}$ = 4.6, $\underline{SD}$ = 3.04). The minimum amount of experience reported was one and a half years, while the maximum amount was 10 years.

Figure 1: Mean previous search experience on different systems reported by subjects.



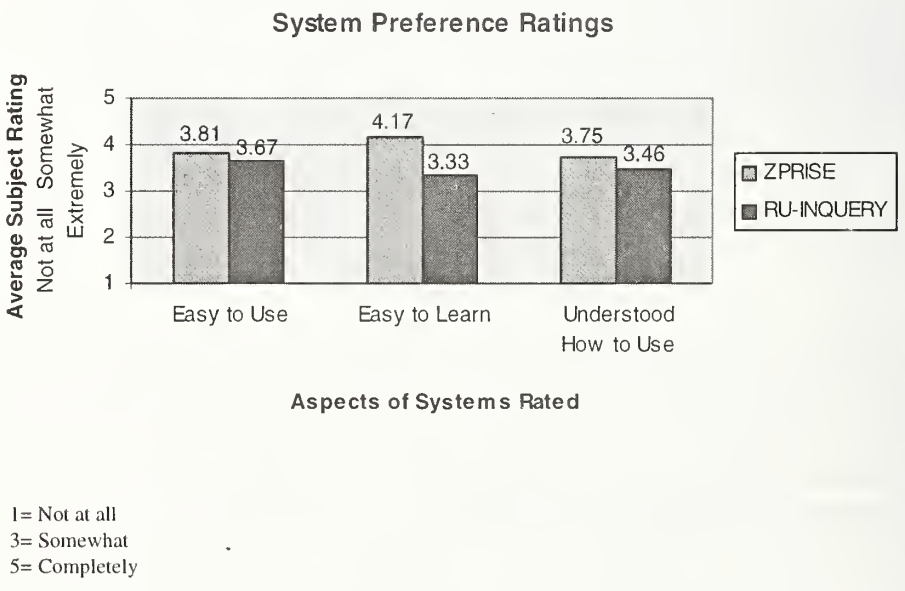
As can be seen in Figure 1, the average amount of previous search experience on different types of systems varied widely. Using a five point scale ranging from none to a great deal of experience, the most experience was reported using mouse-based interfaces ($\underline{M}$ = 4.5, $\underline{SD}$ = 1.07). Relevance feedback had the lowest reported experience ratings ($\underline{M}$ = 2.25, $\underline{SD}$ = 1.28). Surprisingly, the average rating for experience on commercial online systems, such as Dialog, Lexis and BRS Afterdark, was fairly low ($\underline{M}$ = 2.38. $\underline{SD}$ = 1.19). Otherwise, the subjects reported having a fair amount of experience on each of the different system types (ranked output systems, $\underline{M}$ = 2.87, $\underline{SD}$ = 1.25; CD ROM, $\underline{M}$ = 3.38, $\underline{SD}$ = .52; full-text databases, $\underline{M}$ = 3.38, $\underline{SD}$ = 1.06; WWW search engines, $\underline{M}$ = 3.75, $\underline{SD}$ = 1.28; library catalogs, $\underline{M}$ = 4.0, $\underline{SD}$ = .76).

3.2 Subjective Ratings of Searchers

After each search the subjects provided subjective ratings related to the specific search and to the system they used. These ratings were made based on a 5-point scale where 1 was "not at all," 3 was "somewhat" and 5 was "extremely." Collapsing across topics and systems, the 8 subjects felt they were mildly to moderately familiar with the topics (M = 2.46, SD = .56), that the searches on the topics were moderately easy (M = 3.42, SD = .70), that they were somewhat satisfied with the results (M = 3.06, SD = .88) and somewhat confident that they identified all the possible aspects for the topic (M = 2.92, SD = .94). There was more variability on responses to whether subjects felt they had sufficient time to do an effective search, although it was moderately high (M = 3.60, SD = 1.27).

Although there were significant correlations between each pair of subjective performance measures, there was no significant correlation between rated familiarity with the search topic and confidence with the search, ease of searching, satisfaction with the search nor with sufficiency of time for the search (r_{pb} = -.04, ns; r_{pb} = .08, ns; r_{pb} = -.007, ns; r_{pb} = .002, ns, respectively). This supports the assumption that variability in subject familiarity with the search topics should not strongly impact the findings of the study. System order was also evaluated by comparing the average responses on the subjective performance measures from the first system used to the second system used. No significant differences were identified (ease of search, t (7)= -.92, ns; confidence in search results, t (7) = -.50, ns; satisfaction with search, t (7) = -.92, ns; sufficient time for search, t (7) = -1.52, ns). The order in which the subjects used the systems did not significantly influence their subjective ratings of their search performance.

Figure 2: Subject ratings of ZPRISE and RU-INQUERY across search topic on ease of use, learning and understandability. (Note: N = 8)

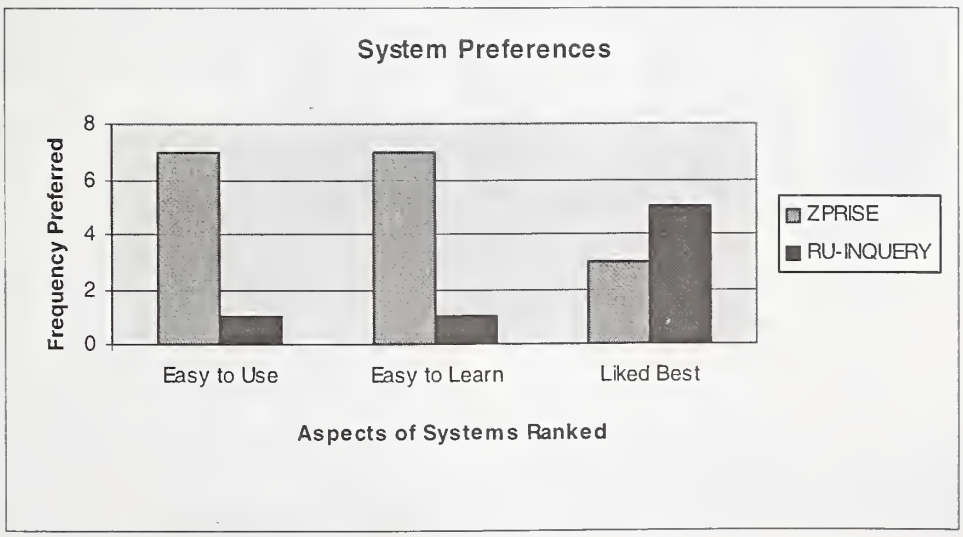


Overall, the subjects rated both systems above average on ease of use, learning and ability to understand how to use it (see Figure 2). The five-point scale used "not at all," "somewhat" and

"extremely" as anchors. The average rating on each of these areas was higher for ZPRISE than for RU-INQUERY. Indeed, the highest average rating for RU-INQUERY is still lower than the lowest average rating for ZPRISE. When ratings of the RU-INQUERY systems are evaluated separately, it is clear that the 4 subjects using both positive and negative relevance feedback rated the system higher and more consistently (easy to use, M = 4.0, SD = .72; easy to learn, M = 3.67, SD = .27; understand how to use, M = 3.67, SD = .98) than those only receiving positive relevance feedback (easy to use, M = 3.33, SD = 1.19; easy to learn, M = 3.0, SD = 1.27; understand how to use, M = 3.25, SD = 1.28). When comparing the average ratings of the 4 subjects using the higher ranking RU-INQUERY to the average ratings of the 8 subjects on ZPRISE, RU-INQUERY has a slightly higher average rating for ease of use, but remains lower on ease of learning and understanding how to use it.

Subjects provided additional subjective ratings, relative to their overall experience, after doing all 6 searches on the two systems. On a five point scale where 1 is "not at all", 3 is "somewhat" and 5 is "completely", on average, subjects rated their understanding of the task very highly (M = 4.0, SD = 1.07). They rated the search tasks in the study moderately similar to searching tasks they typically perform (M = 3.5, SD = .93). They rated ZPRISE as somewhat different compared to the RU-INQUERY system that they worked on in the study (M = 3.38, SD = .74). When comparing ZPRISE to the RU-INQUERY system on ease of use, 7 of the 8 subjects identified ZPRISE as easier to use. The one subject choosing RU-INQUERY as the easier system was using the version with only positive relevance feedback. Similarly, 7 of the eight subjects identified ZPRISE as the system easier to learn and again the one subject choosing RU-INQUERY had no negative relevance feedback. Interestingly, however, even with the preponderance of subjects identifying ZPRISE as easier to learn and use, only 3 of the 8 subjects selected ZPRISE as the system they liked best. This is illustrated in Figure 3. Three of the four subjects using the positive relevance feedback only version and 2 of the 4 subjects using the version with both negative and positive relevance feedback selected RU-INQUERY as the best system compared to ZPRISE.

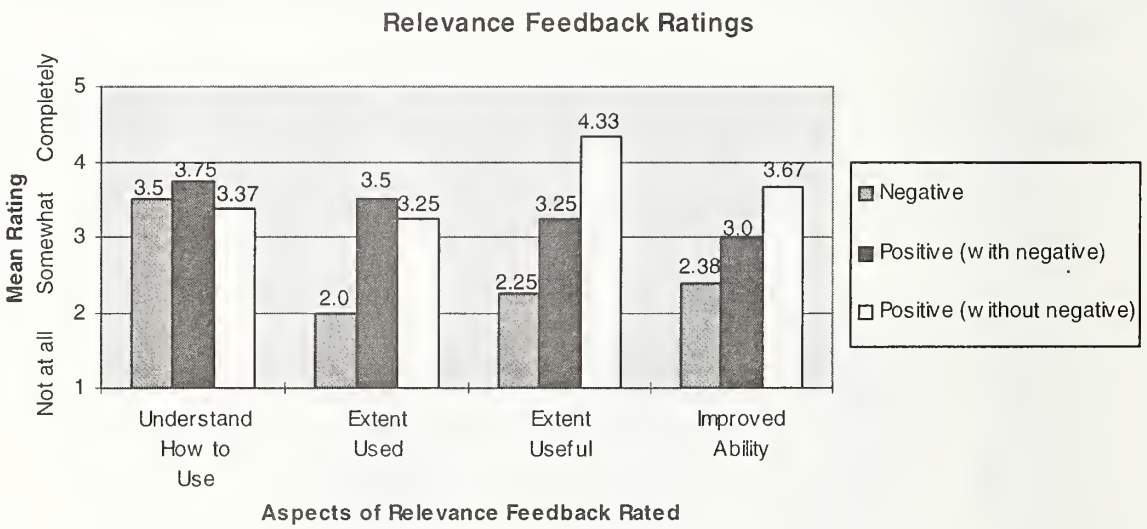
Figure 3: The frequency with which subjects preferred ZPRISE or RU-INQUERY on each of three system aspects.



Subjects provided comments about the two systems, during the exit interview (discussed below in section 3.4). The comments provide a better perspective on their inconsistent ranking of the two systems. It is clear that the subjects felt that RU-INQUERY was powerful and flexible. However, they would forget how to use all the different features or become confused. ZPRISE was seen as simpler and less sophisticated.

Subjects also provided subjective ratings regarding the use of relevance feedback, ranked output and system suggested terms. Subjects responded on a 5-point scale (1 = "not at all," 3 = "somewhat," 5 = "completely"), to questions exploring the extent to which relevance feedback was understood, used, found useful and thought to improve search abilities. The subjects using the version of RU-INQUERY with both positive and negative relevance feedback, provided a response for each type of relevance feedback. The mean ratings for each type of relevance feedback can be seen in Figure 4. The average rating for understanding how to use relevance feedback was above the midpoint of the scale. However, the variance in responses was much greater for subjects only rating positive relevance feedback (positive only SD =1.79, positive SD = .96, negative SD = .58). Subjects were less likely to use negative than positive relevance feedback. When negative relevance feedback was not available, positive relevance feedback was generally rated higher for usefulness and as improving ability to identify different aspects of the topic. Negative relevance feedback was not reported to be very useful or considered to improve searchers' ability. Using the same scale, subjects rated the extent to which ranked output and system suggested terms were useful. Generally, subjects found both features to be moderately to highly useful (ranked output, M = 4.13, SD = .84; system terms, M = 3.71, SD = 1.11).

Figure 4: Average subject ratings on different aspects of subjective responses to using relevance feedback. (Note: n = 4).



3.3 Characteristics of the Interactions

We are primarily concerned with differences in measures of interaction between ruinq1 and

ruinq2 which indicate different aspects of usability of the systems. There appears to be no significant difference in the time taken to learn to use the two systems (ruinq1 tutorial mean of 1612.25 seconds, ruinq2 tutorial mean of 1593.5 seconds). Nor is there any significant difference in the time taken per search (ruinq1 mean of 1096.25 seconds, ruinq2 mean of 1192.08 seconds), which seems to indicate that they are equally easy (or difficult) to use. However, differences do arise in other measures of the interaction.

The numbers of iterations, or cycles, per search are quite different (mean for ruinq1 8.92, mean for ruinq2, 5.17), which means that although the total time is more or less the same for both systems, time per cycle is greater for ruinq2. This may also be related to the large difference between the two systems in the numbers of full texts viewed (ruinq1 mean of 25.17, ruinq2 mean of 52.17), and in titles viewed (ruinq1 mean of 378.08, ruinq2 mean of 205.25), which suggests that searchers in the ruinq2 condition spent much more time reading texts than those in the ruinq1 condition, while those in the latter spent more time scrollling through the retrieved document list. There seems to be no obvious relationship between the different features of the two systems, and these differences in behavior, although futher analysis, particularly of the thinking aloud data, may help to explain them. The searchers in the ruinq2 condition made more use of relevance feedback terms (ruinq1 mean of 7.42, ruinq2 mean for positive terms of 11.08, and for negative terms of 4), which effect is heightened since there seems to be no great difference in the number of positively marked documents in the two conditions (ruinq1 mean of 4.25, ruinq2 mean of 5).

Overall, on these quantitative measures of the interaction, although there are some evident differences between behavior in the two systems, they are not easily explained by the presence or absence of support for negative RF, and may be the result of searcher differences rather than system differences.

3.4 Exit Interview Data

During the exit interview, searchers discussed their experiences using relevance feedback. Almost all of the subjects said they understood how to use relevance feedback, at least to some extent. However, as mentioned above, subjects were less likely to use negative than positive relevance feedback; and when negative relevance feedback was not available, positive relevance feedback was generally rated higher for usefulness and for improving ability to identify different aspects of the topic, than negative RF when it was available.

The following are some of the reasons searchers gave for finding *positive relevance* feedback helpful:

1.Positive RF Helps to Identify Relevant Terms and Aspects

For the aspect task, searchers were required to identify as many aspects of the topic as possible. Positive feedback reportedly helped to ensure that all of the relevant terms had been covered. As one searcher told us during the interview, "It helps me like a thesaurus would help me to make sure I'm covering all different terms." (S002)

2. Positive RF Helps to Assist Learning

Positive relevance feedback appears to be helpful in assisting searchers to think not only about what is missing in the search, but also about what he or she is doing in the search process. For example, "There were things as we went through relevant articles, there were things that I that I

just didn't think of at first and its almost like brainstorming. It sort of prompted me to think a little more about exactly what I was doing, and not only did I use them by adding them into my query, it also helped me by knowing what I didn't want to add to my query as well." (S006)

3. Positive RF Helps to Save Time

For this task, searchers were required to finish each search within 20 minutes. Positive relevance feedback allowed them to identify the relevant documents without reading the whole article. For example: "...it allows you to zero in quickly on the ones that would be useful without having to read through the article." (S001)

4. Positive RF Reduces the Retrieved Set Size

One of the reasons that searchers in our study liked positive feedback was that it seemed to keep the set of retrieved documents smaller, thereby making searching more efficient. The following searcher expressed this feeling: "That was useful cause it keeps the pile getting smaller." (S007)

Our exit interview data reveal several reasons why *negative relevance* feedback was difficult to use, or was *not* helpful:

1. Negative RF Sorts Out Relevant or Partially Relevant Documents

The major concern that our searchers had about using negative feedback was that it might sort out, or eliminate, some articles they would want to look at. For example, "The only problem is that its kind of difficult, because some of the articles you want partially, but you say you already have something similar and you don't want anything else to do with that specific topic. You really can't put negative on it because then it might sort out some other articles that you may want." (S003)

2. Negative RF Reduces the Rank Position of Relevant Documents

Another concern about using negative relevance expressed by our searchers was that it might push back relevant documents on the ranked list. As this searcher told us, negative RF was not helpful "because it pushes them back, and maybe those are articles you wanted..." (S003)

3. Negative RF is Difficult to Use Under Pressure

Some of our subjects found negative relevance feedback difficult to use because of the time pressures imposed by the experimental conditions. In other words, using negative RF takes time and a relaxed searching atmosphere. For example, "Actually, for example, if I get used to this search engine, I'll use positive and negative feedback , but now I am a participant, and I got this feeling that I have to do good. Maybe it's just like I'm in a test. Maybe I have too much pressure. Because I was afraid to wonder around, play around. I feel restricted, that I have to complete this task." (S008)

4. The Usefulness of Negative RF is Topic Related

According to searchers, the usefulness of negative RF varies by search topic. Some of the topics are quite straightforward, in which case there is no perceived need to use negative feedback. As this searcher put it, "I just used it to try to get a word...A lot of the searches
I had were straightforward so I didn't need to." (S004)

5. Word Stemming is Problematic in Negative RF

Some searchers thought that word stemming made it difficult to use negative (and positive) relevance feedback effectively. "The negative (RF) and I think positive, too, they only go by word stem, so I got a lot of things about universities, and when I tried to use negative on it, it'll show up on tape, universe, you could use negative on that stem, and then you'd be throwing out things. So maybe negative and positive shouldn't have a stem." (S004)

6. Negative RF is Simply Disliked

Some searchers just did not like negative relevance feedback, for unexplained reasons, so they did not even try to use this function. For example, "I didn't really. I didn't like it the first time. I didn't bother with it." (S007)

7. Negative RF Does Not Offer Term Control

One suggestion from searchers is that they would like to be able to type their own words, and this would make negative RF more effective. For example, "It would be really cool if you could type in, actually type in, what words you didn't want, or what words were cool, other than just putting them in the key word thing." (S004)

3.5 Performance Results

Because of the technical problems we experienced in logging our searches, which we have described above, we are not able to present comparative results between experimental and control systems. Instead, we discuss differences in performance outcomes on the aspect recall task for the two versions of our experimental system, one with positive RF only and the other with both positive and negative RF.

Our original hypothesis was that the system with both positive and negative relevance feedback will lead to better search performance than the system with positive relevance feedback only, and users will prefer negative relevance feedback to positive relevance feedback. This assumption is drawn from our analysis of thinking-aloud protocols, interviews, and questionnaires from TREC4 and TREC5 data.

Tables 1 and 2 summarize the descriptive statistics of precision and aspect recall between our two different systems: ruinq1 indicates the system with positive relevance feedback only, and ruinq2 with positive and negative relevance feedback.

Table 1. Average Precision for Searches on ruinq1 and ruinq2 (N=24)

System	M	SD	Min	Max
ruinq1	.67	.35	.00	1.00
ruinq2	.66	.22	.33	1.00

Table 2.Average Aspect Recall for Searches on ruinq1 and ruinq2 (N=24)

System	M	SD	Min	Max
ruinq1	.46	.37	.00	1.00
ruinq2	.53	.33	.11	1.00

The mean precision of ruinq1 ($\underline{M}$ = .67, $\underline{SD}$ = .35) and ruinq2 ($\underline{M}$ = .66, $\underline{SD}$ = .22) are almost the same. The result of an independent samples t-test also indicates that there is no significant difference in precision between ruinq1 (INQUERY with positive relevance feedback only) and ruinq2 (INQUERY with both positive and negative relevance feedback).

In this experiment, we were more interested in the measure of "aspect recall" than "precision", because the focus of the searchers' task was on the identification of as many aspects of the specific topic as possible. The mean aspect recall of ruinq2 ($\underline{M}$ = .53, $\underline{SD}$ = .33) is higher than the mean aspect recall of ruinq1 ($\underline{M}$ = .46, $\underline{SD}$ = .37). Contrary to our initial expectation, there is no significant difference in aspect recall between ruinq1 and ruinq2. This insignificant result is partly a result of there being too few subjects for analysis (we had only four subjects for each system). However, the comparison is in the expected direction: the system with both negative and positive relevance feedback leads to better performance than the system with only positive relevance feedback. A replication of this study with a larger sample size, or different sampling method, might reveal significant differences in performance between these two different systems. This remains an open area of investigation.

We were also interested in the possible relationships between demographic characteristics of the searchers and their performance, and also in the relationships between their subjective evaluation of their searches and their actual performance. Contrary to our expectation, none of the demographic or experience variables obtained from the pre-search questionnaire is significantly related to performance measures (aspect recall and precision). Also, there is no significant relationship between searchers' subjective evaluations and their actual performance. Again this result can be partly explained by small sample size.

3.5 System Comparisons

As a final step in our analysis we compare performance measures of all of the participating systems in the TREC6 Interactive Track. We compare the average recall and precision of all the participant systems (except for Rutgers and UMASS's INQ4int, which did not provide results of the common control system). We find that although there is a positive correlation on recall between experimental minus control systems and experimental systems (E-C vs. E, $r = .84$, $p < .001$), the recall of experimental systems is also correlated to that of control systems (E vs. C, $r = .57$, $p < .05$). Such results imply that searcher effects are greater than system effects in general at any one site. In other words, searchers at Berkeley had better recall performance than other sites in terms of both experimental and control systems. A comparison of characteristics of searchers at different sites may provide explanations for searcher effects. Secondly, the same thing happened to precision. While there is a positive correlation in precision between experiment minus control systems and experiment systems (E-C vs. E, $r = .73$, $p < .01$), the precision of experimental systems is also positively correlated to that of control systems (E vs. C, $r = .75$, $p < .01$). Searcher effects were thus dominant in precision. It seems, therefore, that experiment minus control measures appear to be a better indicator for system performance than the measures of the experimental systems alone. This confirms the design of the interactive track for comparison of different experimental systems. Thus, we are unable to fairly compare the performance of our systems with those of other participants.

4. Conclusions

It is difficult to draw any firm conclusions with respect to our initial hypotheses about the benefit of negative RF in the aspectual recall task. Clearly, this is in part a result of the small number of subjects, and perhaps also a result of the lack of a control system correcting for searcher variability. Given the somewhat contradictory nature of the evaluations of the systems

by the subjects in the scale measures as opposed to the free descriptive comments about the system features, and also the fact that ruinq2 performed at least as well as ruinq1, it may be that the most that we can say now is that ruinq2 offered our subjects a useful functionality, implemented in a rather unhelpful way.

Looked at from a slightly more optimistic point of view, it does appear that our results indicate that negative RF, implemented in this way, and subject to the control of the searcher, at the very least does not harm interactive IR performance, and may enhance it. This interpretation is of some interest, since it contradicts previous results using negative RF, especially those in which negative weights have been used. Thus, we tend to consider this study as a promising beginning for more extensive and controlled research on how best to implement and support negative RF in interactive IR.

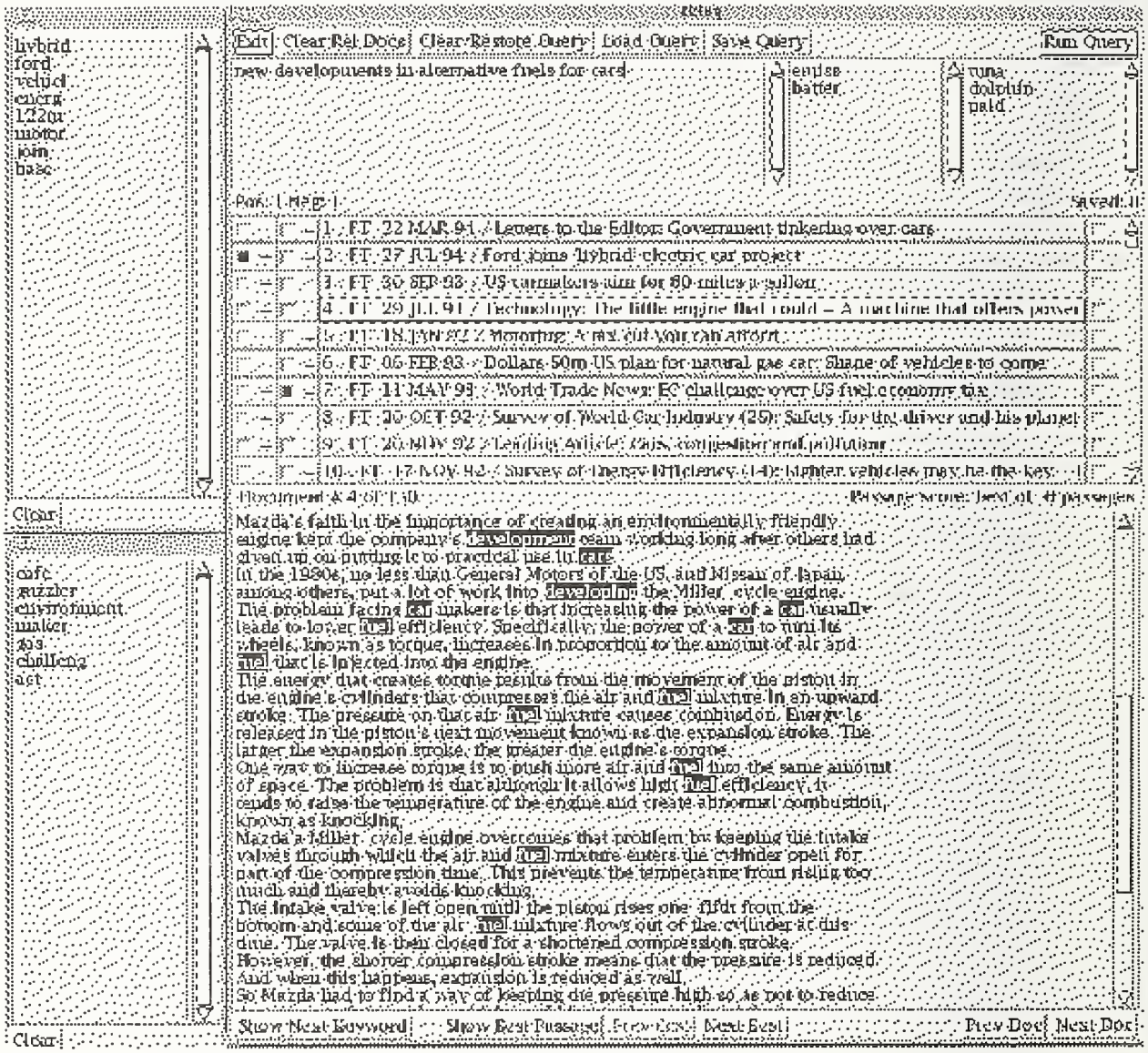
5. References

Belkin, N.J., Cool, C. & Koenemann, J. (1996). On the potential utility of negative relevance feedback in interactive information retrieval. In: SIGIR '96. Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, ACM: 341 (abstract of a poster presentation).

Koenemann, J. (1996) Relevance feedback: Usage, usability, utility. Ph.D. Dissertation, Department of Psychology, Rutgers University.

Koenemann, J. & Belkin, N.J. (1996). A case for interaction: A study of interactive information retrieval behavior and effectiveness. In: CHI '96. Proceedings of the Conference on Human Factors in Computing Systems. New York, ACM: 205-212.

APPENDIX A: SCREEN DUMP OF RUINQ2 INTERFACE



Application of Logical Analysis of Data to the TREC6 Routing Task

Endre Boros
RUTCOR, Rutgers University
boros@rutcor.rutgers.edu

Paul B. Kantor
Jung J. Lee*
Kwong Bor Ng
Di Zhao
Alexandria Project Lab, SCILS, Rutgers University
{kantor, jungjlee, kbng, dizhao}@scils.rutgers.edu

1. The Logical Analysis Approach in the Official Runs

Our approach to TREC6 has explored the possibility of building complex Boolean expressions which represent the classificatory information present in the training data. The positive (i.e. judged relevant), and negative (i.e. judged not relevant) documents are studied separately, using Church's measure of "non-Poissonicity" (Church & Gale, 1995) to identify promising terms for classification.

In the official runs, statistics are produced using the ***MG*** (Witten, Moffat, Bell, 1994)) search engine, and the terms are in fact stems, rather than complete terms. The top 25 terms selected from the positive and negative examples are merged, to form a list with no more than 50 terms. The ***MG*** retrieval system is used (massively) to transform every judged document into a Boolean vector with one component for each distinct classification term. The RUTCOR ***LAD*** program (Boros, Hammer, Ibaraki, Kogan, Mayoraz, & Muchnik, 1996) is used (twice for each topic), with several modifications, to search exhaustively for Boolean prime implicants which characterize the positive and the negative examples. Due to computer speed limitations, we have limited the search in our official submissions to terms of order three (i.e terms such as ABC', where C' denotes the absence of term C). Each pattern which matches some positive (respectively, negative) examples is given a weight determined by the number of examples that it matches.

2 Detailed Procedures of the Official Runs

2.1 Training

For each topic, we used ***MG*** to index all the judged relevant documents to build a index structure, and to compute the term frequencies and document frequencies of all word-stems. We

---

* Permanent address: Department of Statistics, Soong Sil University, Seoul, Korea.

selected 25 word-stems according to the Church criterion (Church & Gale, 1995) on distributions of term frequencies and document frequencies. We did the same for the judged non-relevant documents. For topics with more than 50 Mbytes of judged non relevant documents we randomly selected 50% of the judged documents for MG to index. (Topics: 77, 78, 82, 94, 95, 100, 108, 118, 119, 123, 125, 126, 128, 142, 161, 173, 187, 194, 228, 240, 282). This yields 25 word-stems from relevant documents and 25 word-stems from non relevant documents. Each stem was submitted as a Boolean query, using ***MG***. This produced a list of documents in which the term appeared. These lists are next combined to form a single file in which each relevant document is represented by a single row of 0s and 1s, where 1 signifies that the stem labeling the corresponding column appears at least once in the document. This is the form of case representation accepted by ***LAD***. We do the same for the non-relevant documents, producing a second array of cases.

For each topic, we concatenate the files for the relevant and no-relevant training examples, the degree is set to k, and ***LAD*** finds all Boolean monomials with k literals, matching some relevant document vectors and no non-relevant document vectors. These are the positive patterns of the topic. Negative patterns are defined correspondingly to match non-relevant documents. Thus ***LAD*** provides the foundation for Boolean classification rules.

The process takes time exponential in k. We were limited to $k = 3$ by time constraints. For topic 44, we could not find any positive patterns. We used the patterns file to assign a weight for each pattern, equal to the number of training documents that fit the pattern. Note that due to limitations of person-power and time, our training phase did not contain any evaluation and tuning of the numerous parameters in both attribute selection and ***LAD*** pattern-finding. More details about implementation and algorithms are in section 4.

2.2 Testing

We used ***MG*** to index the test collection of routing documents. We used a stepwise fusion process to produce, from the Boolean patterns, ranked lists. The positive patterns were used to produce a reduced set (except for topic 44). The reduced set is the union of all documents retrieved by any of the patterns. We used two methods to fuse the documents into one single ranked list. The first method is a "quorum" method. Documents are ranked in decreasing order of the number of patterns which retrieved them. The second is weighted fusion. Each pattern has a weight equal to the number of training documents that it covers. The score assigned to a document is the sum of the weights of all the patterns which cover it. Both quorum and weighted scores were also computed for each document, using the set of negative patterns.

Our submission was the top 1000 documents of the positive rank list. We planned to eliminate or re-order those documents retrieved by positive queries according to the ranked list produced by negative pattern queries until the positive document ranked list had only 1000 documents, i.e., we would eliminate from the positive document list the documents that are also in the negative document list and eliminate those with the highest rank of the negative document list first, and so on until the positive document list contains only 1000 document. We found that using this method to eliminate documents could easily eliminate much more than desired number of documents. That is, at certain ranks in the negative list, we would acquire a batch of "knockouts" which brought the remaining list below 1000.

3 Results of the Official Runs

The results are not distinguished. Using the exact averaged precisions our results are occasionally worse than the "worst". We therefore concentrate on the precision at 100 documents. The weighted method performs better than the quorum method in 11 cases, and worse in 8. They are tied in 28. More importantly, the Quorum method produced the worst recorded result in 30 cases, and the weighted method did so in 29 cases. It is clear that the combination of decisions that we have made does not solve the routing problem. We suspect that several factors combine to produce this discouraging result.

4. The LAD Approach in the Non-Official Runs

After we submitted our official runs, we continued our experiments . We have implemented a method based on the Logical Analysis of Data, as it is described in Boros, Hammer, Ibaraki, Kogan, Mayoraz, & Muchnik, 1996 (see for further details, Boros, Hammer, Ibaraki & Kogan, 1997, Boros, Ibaraki & Makino, 1997, and, Boros, Ibaraki & Makino, 1998), with several modifications.

In these experiments, the algorithm we implemented consists of 4 phases. The first phase is a more or less standard indexing of the documents. We have used the ***SMART*** system (version 11.0, implemented on Sun Ultra-1, Solaris 2.5.1), and as a result we have obtained an indexed representation of the documents. Let us denote documents by d, and terms by t, and let us denote by $f(t,d)$ the number of occurrences of term t in document d. The length of a document d is

$$l(d) = \Sigma_t f(t,d)$$

and the relative frequency of a term t in a document d is

$$r(t,d) = \frac{f(t,d)}{l(d)}$$

We have indexed, for all 47 TREC-6 topics, most of the training documents. We exclude IRList digests, Usenet news groups documents, and Virtual World documents. These three collections contain relatively few relevant documents and they are not included in the Tipster document CD collection.

The second phase is a projection, in which we map the high dimensional frequency vector representation into a low dimensional binary representation, essentially following the ideas described in (Boros, Hammer, Ibaraki & Kogan, 1997), with some very important modifications. For a term t and a real number z, let us introduce a propositional statement of the form $X(d)$= "term t occurs with relative frequency higher than z in document d". Such a statement assumes a logical

value (true or false, i.e. 1 or 0) for every document. By choosing such pairs (t_i,z_i) appropriately, and denoting the corresponding propositional variables by X_i for $i = 1,...,k$, we can map every document d into a binary vector $X^d = (X_1(d), ..., X_k(d))$. Ideally, one would like to select pairs (t_i,z_i) such that they "represent well" the particular topic, and one would think, those stating the particular topic are the best to choose. In our algorithm we instead use an automatic learning method for selecting such pairs, and **we did not use the topic descriptions**. For each potential pair (t,z) we computed two parameters:

$$R(t,z) = | \{ d \in RelevantTraining(Topic) : r(t,d) > z + GAP \} |$$

and

$$I(t,z) = | \{ d \in NonrelevantTraining(Topic) : r(t,d) < z - GAP \} |$$

where GAP is a preselected small positive constant. Finally we set $S(t,z) = R(t,z)*I(t,z)$, i.e. $S(t,z)$ counts the pairs of (relevant,non-relevant) documents in the training set of the considered topic, which are "properly" distinguished by the logical statement corresponding to the pair (t,z) with a separation GAP.

In the second phase of the algorithm, we select the smallest set I, indexing pairs (t_i,z_i), for which the separation value $S(t_i,z_i)$ is high and such that for every pair consisting of a relevant document d and a non-relevant document d' in the training set the condition

$$(r(t_i,d) > z_i + GAP) \ AND \ (r(t_i,d') < z_i - GAP)$$

is satisfied by at least M different indices i ($i \in I$), where M is an input parameter. Typically M = 10, or so. In other words, we would like to have a binary encoding of the documents, which is as short as possible, and such that the vectors X^d and $X^{d'}$ are very different, whenever the relevance states of the documents d and d' are different. Since this optimization problem is difficult to solve, we implemented an efficient (polynomial time) approximation to solve it. In our experiments we used $GAP = 0$, $M = 10$, and to decrease the chances of overfitting, we have used only a randomly selected subset (50-80%) of the training documents. The values we obtained for k varied between 30 and 150 for the different topics. Let us add that for most topics just by reading the terms t_i, $i = 1,...,k$, one could get a very good idea of what the topic was about!

In the third phase, we were looking for simple logical rules, in terms of the binary variables X_i, $i = 1,...,k$, characterizing relevance (or non-relevance) well. For instance, if term t_1 is "Japan" and term t_2 is "dump", then X_1 AND X_2 is the simple statement of "the relative frequency of *Japan* is more than z_1 AND the relative frequency of *dump* is more than z_2" and the truth of this for a document d may be a good indicator that d is about "dumping by Japanese companies". More precisely, let us call an elementary conjunction $P = X_{i1} \ AND \ X_{i2} \ AND \ ...$ (in which some variables may appear with a negation) a pattern, if $P(d) = 0$ for all non-relevant documents $d \in Nonrelevant(Topic)$ in the training set, and $P(d) = 1$ for some relevant documents. We say that "the pattern P is triggered" for document d if $P(d) = 1$. In other words, a pattern is a statement which confirms the relevance of some documents, while not raising any false alarms on the non-relevant documents. Let us denote by $C(P)$, called the coverage of P, the number of relevant documents in the training set for which $P = 1$. Obviously, the higher $C(P)$ the more we can trust P (that is, the greater its recall) , and the more we find its existence surprising!

In this phase of our method, we first generate a pattern P for every relevant document d such that $P(d) = 1$, and the coverage $C(P)$ is as high as possible. Since this is again a very hard optimization problem, we employ a fast (polynomial time) approximation algorithm. Let us call the patterns obtained in this stage positive patterns. (Of course, we filter out patterns dominated by others. A pattern is "dominated" if it contains a subpattern which is as effective as it is.) We then interchange the role of relevant and non-relevant documents, and generate a "negative" pattern for every non-relevant document, analogously, i.e., N is a negative pattern if $N(d) = 1$ only for (some) non-relevant documents; and P is a positive pattern if $P(d) = 1$ only for (some) relevant documents.

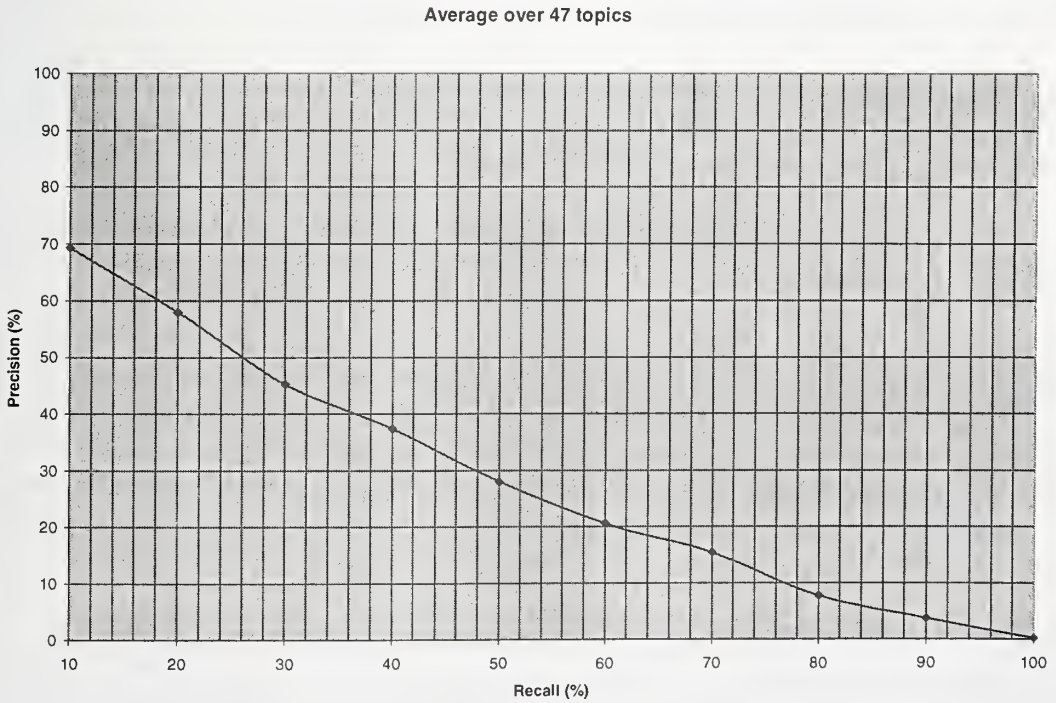
Finally, we select a smallest subset of these positive and negative patterns such that for every document d in the training set at least N of these selected patterns are triggered. (These triggered patterns will be only positive patterns if d is relevant, and must all be negative patterns if d is non-relevant.) In our experiments we choose $N = 5$.

In the last phase we compute a score for every document in the test set by setting

$$S(d) = \sum_{P \text{ positive}} P(d) - \sum_{N \text{ negative}} N(d).$$

Intuitively, if the training set represent the given topic well, this score must be positive for relevant documents, and negative for non-relevant documents.

In our experiments the average precision over all 47 topics was about 28%, which is about the same as the median of all other methods at the TREC-6 meeting. Out of the 47 topics our average precision was better than the median 27 times, and was below the median 20 times.



5. Discussion

This effort represents a first step towards automating the routing process in a way which reflects the natural human preference for (and documented effectiveness of) Boolean formulations as a basis for ad hoc retrieval. There are many ways in which the present first attempt can be expanded, including the search for effective synonyms appearing in the Boolean implicants, and more subtle methods for combining the patterns found. As an example, since all the judged documents in the TREC setting have been retrieved by other systems in prior years, one might have an "Inclusion rule" built on all of the judged documents (contrasted to a random selection of unjudged documents), followed by an "Exclusion step" based on patterns which resolve the judged relevant from the judged non-relevant. Conceptually, this approach is based on the belief that Boolean combinations of "terms" which are in turn surrogates for "concepts" are a powerful representation of texts when the goal is to estimate relevance.

While it is customary to treat the routing task as deriving most of its information from the judged documents, we intend to examine this assumption, in our setting, by looking next at the terms of the topic, to see whether there are any potential useful terms that were not discovered by our process. If there seem to be any such terms we will test what happens when they are added to the set of basic variables. Since many TREC systems are vector based, we conjecture that this effect is most likely to occur if a Topic specified that the document is to be "not about such and such".

Many choices were made in the press of time, and without any systematic evaluation of the alternatives. It is our present belief that the following factors may explain the improvement between our official runs and the subsequent experiments reported here.

1. Originally the stems forming the basis for binarization of the data were chosen on a distributional criterion. Now they are chosen on the basis of power in separating the positive and negative instances in the training set.

2. Our choice of degree 3 was due to resource limitations. The current method finds some patterns of very high degree.

3. Our training set consists of documents which were retrieved in prior years, by systems which behave in roughly similar fashions. Thus our training procedure may not be the most logical one. An alternative is a two-step procedure: (1) find patterns which distinguish retrieved documents from all documents; (2) find patterns which distinguish the non-relevant retrieved documents from the relevant ones. This alternative procedure corresponds more faithfully to the way in which the patterns were used in our official submission, and it seem reasonable that training towards this purpose will produce better results.

Formally, the ***LAD*** method, as opposed to vector classifiers, or even quadratic classifiers, supports retrieval of substantially distinct clusters of relevant documents, in the underlying vector space with word stems for axes. This is, in principle, attractive, as it exploits a special feature of

normal human searching. However, the present results show that the methods for finding the clusters our patterns must be made substantially more powerful to be competitive with todays state-of-the-art vector based retrieval systems.

Acknowledgements : This work has benefited enormously from conversations with Peter L. Hammer, Alex Kogan, Slava Brover, and Ken Church.

References

Boros, E, Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E. and Muchnik, I. (1996). An Implementation of Logical Analysis of Data. Rutcor Research Report 22-96 RUTCOR, Rutgers University.

Boros, E, Hammer, P.L., Ibaraki, T., & Kogan, A. (1997).Logical analysis of numerical data. Mathematical Programming, 79, 163-190.

Boros, E., T. Ibaraki, T, & Makino, K.(1998) Error-free and best-fit extensions of partially defined Boolean functions.Information and Computation, 140 (2), 254-283.

Boros, E., Ibaraki, T., & Makino, K. (1997). Monotone extensions of Boolean data sets, in: Algorithmic Learning Theory -- ALT'97 (M. Li and A. Maruoka, eds.).Lecture Notes in Artificial Intelligence 1316, Springer, pp.161-175.

Church, K. F. & Gale, W. A. (1995). Inverse Document Frequency (IDF): A measure of deviation from Poisson. Proceedings of the Third Workshop on Very Large Corpora.

Witten, I. H, Moffat A, Bell TC. (1994). Managing Gigabytes. New York: van Nostrand Reinhold.

The text categorization system TEKLIS at TREC-6

Thomas Brückner

Siemens AG
Otto-Hahn-Ring 6
81730 München, Germany
thomas.brueckner@mchp.siemens.de

Abstract

This short paper documents our participation on the filtering and routing tasks of TREC-6 with the commercial filtering system TEKLIS. TEKLIS is a training-based statistical categorization system which incorporates shallow linguistic processing and fuzzy-set methods.

In the following we will present the core technology of TEKLIS, our results on the filtering and routing tasks and a discussion of the insights we gained through our participation.

1 Introduction

TEKLIS is a text categorization system aimed at a wide variety of applications like news indexing, user profile based filtering or hierarchical cataloguing of WWW sites. Since it is a true filtering system it is obviously well suited for the filtering task of TREC but it can be easily used for routing too.

As a first time participant we concentrated on getting our system to run with the TREC data and instead of making various experiments made a thorough analysis of test runs on older TREC data to see what went possibly wrong and why. We will present the results in the discussion section.

2 System Overwiew

In the first processing step of a text TEKLIS normalizes the words with a lemmatizer. The lemmatizer returns for each word one or more stems and it's lexical category. A statistical HMM Part-of-Speech Tagger resolves lexical ambiguities.

Categorizing a text in our system depends on the relevance of words for categories. The relevance of a word for various given categories is computed from a set of training texts. We define the relevance of a word w for a category c, rlv(w in c), through Pearsons correlation coefficent $r(w,c)$. Negative relevances are not used, since they usually worsened the results in practice.

If we categorize a new text we get for each category c a set C of relevances for each word w in text. Now we can think of the set C as a fuzzy set with membership function $\mu_C(w) = rlv(\text{w in c})$. For such a set C we compute it's probability with

$$\text{prob}(C) := \Sigma\, \mu_C(w) \,.\, p(w),$$

where $p(w)$ is interpreted as $p(c|w)$. When the probability for a category c, normalized over the number of words in the text, is higher than a given threshold, the system returns the category. By varying the threshold it is possible to get different recall/precision levels (higher threshold = lower recall and higher precision, lower threshold = higher recall and lower precision).

For a more detailed description of the system see [1].

3 Filtering task

Since TEKLIS is a true filtering system we only had to create an index file for the training data and changed the output to accomodate the TREC format. Otherwise we could run the system „as is" on the filtering training and test data. As TEKLIS is designed to learn from sample documents we didn't used the topics descriptions, which might be a disadvantage (see Discussion). For predicting which thresholds are best suited for the different runs (F1, F2, ASP) we made some tests on older TREC data.

Our results on the filtering task were as shown in the table below:

Topic	#relevant	F1 (run 1)	F1 (run 2)	F2 (run 1)	F2 (run 2)	ASP (run 1)	ASP (run 2)
1	51	1	2	-43	-46	0.025	0.018
3	76	0	0	-76	-76	0.000	0.003
4	80	0	0	-80	-80	0.000	0.000
5	7	-48	-68	-61	-82	0.000	0.002
6	165	54	68	-28	-2	0.150	0.169
11	174	-2	-8	-151	-124	0.023	0.056
12	292	0	0	-269	-200	0.014	0.065
23	7	0	0	-8	-8	0.000	0.000
24	42	1	1	-43	-45	0.004	0.003
44	4	0	-2	-5	-5	0.000	0.000
54	174	0	0	-174	-175	0.000	0.000
58	18	0	0	-18	-18	0.000	0.000
62	401	-19	-35	-418	-435	0.001	0.001
77	16	-2	-2	-17	-18	0.000	0.000
78	45	0	0	-45	-45	0.000	0.000
82	82	0	-2	-85	-88	0.000	0.000
94	193	-13	-38	-217	-219	0.001	0.002
95	138	0	-8	-148	-160	0.000	0.000
100	197	0	0	-197	-197	0.000	0.000
108	314	0	0	-314	-311	0.000	0.002
111	566	-40	-70	-556	-556	0.000	0.000
112	59	0	0	-59	-59	0.000	0.000
118	59	-32	-160	-154	-154	0.006	0.006
119	85	-6	-11	-85	-102	0.009	0.007
123	62	0	0	-63	-64	0.000	0.000
126	27	-28	-38	-50	-56	0.000	0.000
126	19	-16	-18	-30	-37	0.027	0.022
126	333	0	0	-333	-325	0.000	0.006
142	229	-30	-16	-139	-139	0.063	0.063
118	260	-20	-23	-268	-286	0.003	0.002
154	175	-2	-2	-174	-174	0.000	0.004
161	121	15	18	-72	-46	0.027	0.132
173	16	-12	-22	-38	-12	0.000	0.000
180	17	-6	-8	-28	-22	0.000	0.009
185	18	-2	-2	-18	-15	0.000	0.028
187	21	-2	-2	-22	-25	0.000	0.000
189	890	0	0	-886	-874	0.001	0.004
192	7	0	0	-8	-11	0.000	0.000
194	4	-6	-12	-13	-18	0.000	0.000
202	627	-15	100	-427	-427	0.057	0.057
228	65	0	0	-66	-71	0.000	0.000
240	131	0	0	-131	-128	0.000	0.004
282	28	-6	-10	-38	-43	0.000	0.000
10001	135	-2	-6	-141	-142	0.000	0.000
10002	321	-3	-5	-326	-325	0.000	0.001
10003	73	-10	-12	-83	-88	0.000	0.000
10004	18	-38	-44	-50	-66	0.000	0.000

TREC-6 filtering results for TEKLIS

4 Routing task

For applying TEKLIS to the routing task of TREC we used the normalized fuzzy set probability described in chapter 2 as the ranking function. Otherwise we didn't changed the algorithm.

Our results on the routing task (category B) were as shown below:

Total number of documents over all queries
  Retrieved: 46000
  Relevant: 3499
  Rel_ret: 2580

Interpolated Recall - Precision Averages:
  at 0.00 0.4988
  at 0.10 0.3484
  at 0.20 0.2765
  at 0.30 0.2388
  at 0.40 0.1990
  at 0.50 0.1682
  at 0.60 0.1497
  at 0.70 0.1243
  at 0.80 0.0837
  at 0.90 0.0472
  at 1.00 0.0101
Average precision (non-interpolated) over all rel docs:
  0.1774

Precision:
  At 5 docs: 0.3000
  At 10 docs: 0.3022
  At 15 docs: 0.2884
  At 20 docs: 0.2728
  At 30 docs: 0.2652
  At 100 docs: 0.2115
  At 200 docs: 0.1577
  At 500 docs: 0.0920
  At 1000 docs: 0.0561
R-Precision (precision after R (= num_rel for a query) docs retrieved):
  Exact: 0.2053

5 Discussion

As mentioned earlier we tested our system on older TREC data with known relevance judgements. An analysis of the classification errors yielded some interesting results giving important hints for future improvements of TEKLIS:

- Computing a score normalized over the text length is not suitable for TREC data. We found many samples of very long documents, where the relevant passage was only one paragraph. In general the normalized score used by TEKLIS is only good if the documents are short or if a document should be categorized as a whole.

- For specific categories like the TREC topics it is easier to learn „very" relevant words (comparable to concept words) than for more general categories, like economy, sports, politics, etc. But the ambiguity of these words can be greater for categorization purposes.

- The restriction of using single words without context is not sufficient for the TREC topics. We think it's neccessary to enhance TEKLIS with the possibility of learning contexts for the most relevant words. First experiments showed an improvement of 10% in recall and precision on the filtering task.

Finally we tried to incorporate the topic descriptions into our training. Since TEKLIS is designed to learn from sample documents we didn't used the topic descriptions for our official runs. In an experiment we used the topic descriptions as „artificial" documents multiplying them 30 times for each category. After adding these „artificial" samples to the training data we got significant improvements in our test results.

Reference

[1] T. Brückner, P. Suda, H.-U. Block and G. Maderlechner, „In-house Mail Distribution by Automatic Address and Content Interpretation", 5th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, (1996), pp 72-74.

ETH TREC-6: Routing, Chinese, Cross-Language and Spoken Document Retrieval

Bojidar Mateev, Eugen Munteanu,
Páraic Sheridan, Martin Wechsler, Peter Schäuble

Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich, Switzerland

Abstract

ETH Zurich's participation in TREC-6 consists of experiments in the main routing task, both manual and automatic runs in the Chinese retrieval track, cross-language retrieval in each of German, French and English as part of the new cross-language retrieval track, and experiments in speech recognition and retrieval under the new spoken document retrieval track. This year our routing experiments focused on the improvement of the feature selection strategy, on query expansion using similarity thesauri, on the grouping of features and on the combination of different retrieval methods. For Chinese retrieval we continued to rely on character bi-grams for indexing instead of attempting to segment and identify individual words, and we introduced a new manually-constructed stopword list consisting of almost 1,000 Chinese words. Experiments in cross-language retrieval focused heavily on our approach using multilingual similarity thesauri but also included several runs using machine translation technology. Finally, for the spoken document retrieval track our work included the development of a simple speaker-independent phoneme recogniser and some innovations in our probabilistic retrieval functions to compensate for speech recognition errors.

1 Introduction

The introduction of two new tracks to the TREC-6 evaluation have helped ETH Zurich greatly in evaluating the research in spoken document retrieval (e.g. [Wechsler and Schäuble, 1995]) and cross-language information retrieval (e.g. [Sheridan and Ballerini, 1996]) that we have been conducting for the past number of years. We have therefore participated fully in both of these tracks

and, even based on the preliminary results released before the conference, have learned a great deal from the experience. Additionally to these two new tracks, TREC-6 sees our first official submissions to the Chinese retrieval track. These include both automatic and manual runs and build upon work we completed just after the deadline for submissions in last year's evaluation. We also continue to refine our approach to the routing task in the search for performance improvements. Our routing submissions therefore constitute the investigation of several sources of possible improvement over our TREC-5 submissions.

One aspect of our approach to the routing task that we specifically examined for improvements was the use of the U-measure for feature selection [Ballerini et al., 1996]. Further endeavours for the routing task centred on examining the grouping of semantically related features, the use of similarity thesauri and the combination of different retrieval methods. More details of our routing work and experiments are presented in section 2.

For our work on Chinese retrieval we already had in place an indexing module which indexed Chinese texts using character bi-grams. For our initial experiments last year we simply indexed everything using bi-grams and used our standard SPIDER retrieval functions, without any consideration of Chinese language characteristics. We have now included a manually constructed stopword list for Chinese, consisting of almost 1,000 words. Not only did we remove the stopwords from our index, we also used stopwords as obvious word boundaries, indexing only character strings between stopwords. Our Chinese indexing module additionally recognises English words in Chinese text and maintains these as complete units instead of breaking them into bi-grams. Details of our automatic and manual Chinese experiments are presented in section 3.

The cross-language retrieval track provided document collections in each of English, French and German and topic descriptions also in each language. Our work for this track covered all of these languages for the monolingual runs and also English-to-German, German-to-English, German-to-French, and French-to-German cross-language runs. We did not work with French-English cross-language combinations. Most of our cross-language work was based on further examination of our approach using similarity thesauri, especially with respect to the construction of similarity thesauri over bi-lingual corpora of varying quality. We did however perform additional runs using machine translation technology to automatically translation queries between English and German. More details are given in section 4.

We participated in the full Spoken Document Retrieval (SDR) track with our probabilistic weighting approach based on matching documents and queries at the phoneme level. We also built a simple speaker-independent phoneme recogniser using the HTK Toolkit [Young et al., 1993]. Much of our work for this track was centred on refining the probability estimation module used in our approach to retrieval from errorful documents (in this case, the output from speech recognition) and the weighting function used in ranking documents. This is described more fully in section 5.

2 Routing Experiments

In our TREC-5 paper we introduced a method for feature selection based on the U-measure [Ballerini et al., 1996]. The features with the highest value of μ_U were selected and included in the query profile for each topic. This year, for each query the 300 one-word features and the 300 phrases with the highest value of μ_U were selected. Additionally, if a query had a title part then the one-word features from the title are automatically selected and included at the top of the features selected by μ_U value. The selection of features with this method was then used as a source of information for grouping features.

Our aim in grouping features was to gather together *related* one-word features from a pre-selected set. Features can be considered to be related in many different ways for this task. Our approach is inspired by the idea of a *lexical space* as described in [Zavrel and Veenstra, 1995], though we aim to group together semantically related features instead of the syntactic classes described therein. The method depends on two sets of one-word features: a set of *focus* features and a set of *context* features. For our routing experiments we use query-dependent selections for the two features sets, making use of the feature selection method described above. Table 1 shows that using query-dependent feature sets in the creation of lexical spaces (feature groupings) indeed has the desired result of generating query-specific feature groupings. This example shows the feature groups generated around the feature *"dump"* which occurs in three queries.

Query Nr.	5	12	10001
Query	Dumping charges against Japan	Water Pollution	Soil Pollution
Dump...	antidump	discharg	dispos
	penalti	water	discharg
	punit	cleanup	mar
	impos	untreat	widespread
	price	pollut	pollut

Table 1: Features grouped with the feature "dump" for different queries.

Our official submissions for the TREC-6 evaluation used a combination of three different methods for routing retrieval, with the particular parameters of each method adjusted over the training documents from the AP, FBIS, WSJ and ZF32 collections. We used the well-known Lnu.ltn weighting scheme [Singhal et al., 1996] with both one-word and phrasal features selected according the method described above (not grouped). We also employed query expansion using a similarity thesaurus ([Qiu, 1995]) constructed over the FBIS collection. Our second retrieval method for the combination used, for each topic, groups

of features for each of the top 28 one-word features. We computed a ranked document list for each of the 28 groups using the above Lnu.ltn method and then used these lists for each topic as further input to the combination of methods. The third method for the combination was a generalisation of the Binary Independence Retrieval (BIR) model in which we used linear combinations of feature co-occurrence matrices on a per query basis using logistic regression. This method was applied to re-rank initial ranked lists of 2,000 documents which were generated for each query using the Lnu.ltn method.

These three method were then combined on a per-query basis by summing the rank positions of documents in selected ranked lists from those returned by the three different methods (1 ranked list from Lnu.ltn, 28 lists from the feature groupings, and 1 from the BIR method). Selection of ranked lists for each query is based on the training data.

In summary, we use an improved feature selection strategy by adding query title words to features selected by the U-measure. We then group semantically related query features and, treating each group as a query, generate ranked document lists for each group. These ranked lists, and a further ranked list generated by applying the Lnu.ltn weighting method with query expansion, may then be combined with a ranked document list generated from a generalisation of the BIR model using feature co-occurrence matrices. The combination of ranked lists from different methods produces the final ranked list of documents for the query. This combination of ranked lists generated by different methods has been shown to lead to improvements in performance when compared to the performance of the individual methods.

3 Chinese Retrieval

Our work on Chinese text retrieval has been underway since the TREC-5 evaluation, though we did not achieve results in time for the TREC-5 submission deadline. We had simply expanded the SPIDER indexing module to handle Chinese characters and to index Chinese texts using character bi-grams. We then applied this indexing without any further attention to Chinese language considerations - with the sole exception of removing the Chinese two-byte blank-space character when we found that the same document was being retrieved in the top rank position for multiple queries, since matches were taking place on the blank spaces in queries and documents!

Our main extension to our TREC-5 Chinese work is the use of a new manually generated stopword list which consists of almost 1,000 entries. Example entries in the stoplist are illustrated in Figure 1. Apart from using the stoplist to directly eliminate stopword features from the index, we also took advantage of the stopwords to establish obvious word boundaries in the running Chinese text. Our hope is that the text strings between stopwords are in many cases equivalent to phrasal units. In any case, the reduction in number of indexing

features achieved through our stoplist has been substantial. Our TREC-5 index of the Chinese document collection included 2,089,778 individual bi-grams, which has been reduced to 1,148,010 features in our current index, a reduction of 45%. This reduction in the index resulted in no loss of retrieval performance when measured in average precision over the TREC-5 topics.

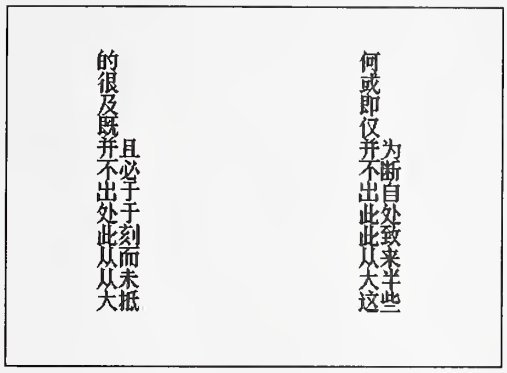


Figure 1: Example entries in our Chinese stoplist

In addition to the removal of stopwords, our Chinese indexing module now also recognises both English words and Chinese numeric strings within Chinese texts and treats these as units for indexing instead of breaking them into bi-grams. The recognition of English words was included in our official TREC-6 submissions whereas the recognition of numbers was developed after our submission. Our automatic Chinese submission used the Lnu.ltn retrieval method with the full topic descriptions as queries and an automatic pseudo relevance feedback loop using the top 10 0 ranked documents to expand the query by the top 50 features.

We also submitted a manual retrieval run in the Chinese retrieval track using two Chinese researchers to search using their own query formulations based on the topic descriptions and their own relevance judgements to be used for relevance feedback. Their task was to find and mark as many relevant documents as possible. The submitted results for each topic were then based on a final retrieval run using the manually identified relevant documents. In searching for relevant documents our first searcher generally read 15 of the retrieved documents, spending approximately 40-50 minutes on each topic. The second searcher generally read about 40 documents for each topic and spent about an hour on each topic. Our searchers identified an average of 11 relevant documents for each topic, though one searcher was noticeably more reluctant to judge documents as definitely relevant than the other (we used more "maybe" relevant documents from that searcher).

4 Cross-Language Information Retrieval

Our participation in the new cross-language retrieval track has covered each of the three individual languages, though not all of the potential language pairings for the cross-language experiments. The main thrust of our work was directed toward the evaluation of our approach to cross-language retrieval which uses multilingual similarity thesauri constructed automatically over comparable corpora. We were also interested however in investigating the potential role of machine translation technology for cross-language retrieval.

Each of our monolingual baseline runs in English, German and French essentially used the same approach. We have constructed stopword lists for each language and also employ stemming or word normalisation for each language. We use Porter's stemming algorithm for English [Porter, 1980], an equivalent rule-based stemmer for French, and a lexicon-based word normalisation module for German which includes compound word analysis. Our German module uses the CELEX lexicon [Baayen et al., 1993]. More details of our indexing for multiple languages can be found in [Wechsler et al., 1997]. Retrieval used the Lnu.ltn retrieval method with the full topic descriptions as queries and an automatic pseudo relevance feedback loop using the top 10 initially ranked documents to expand the query by the top 50 features. Our monolingual runs were denoted ETHee1, ETHff1, and ETHdd1.

4.1 German/French Retrieval

Our German-French cross-language experiments were based on similarity thesauri ([Schäuble, 1997](pp 29), [Sheridan and Ballerini, 1996]). The similarity thesauri were constructed based on the comparable documents of the SDA (Swiss news agency) collections in French and German used in this evaluation. The SDA collection consists of 141,046 French documents and 185,099 German documents. The French and German collections are composed independently and the differing numbers of documents in each language reflect the fact that the German collection reflects much more news local to northern Switzerland and southern German while the French document contain many stories local to southwest Switzerland and southeast France. Despite these differences however, there are many news items common to the two collections, especially relating to international events. It is this commonality which lends the collection its comparability, which we exploit for building similarity thesauri. Comparable documents were identified based on the document dates, co-occurring cognates (words which are identical in each language, especially places and names), and the manually assigned news classification codes assigned by the SDA. Document pairs which scored above a given threshold on matching these features were considered comparable and paired off. This resulted in a collection of 83,698 document pairs considered to be comparable. This resource was then made available to all participants of the cross-language retrieval track.

In constructing German-French similarity thesauri, the 83,698 document pairs were turned into 83,698 "surrogate" bilingual documents by simply concatenating each French and German document of each pair. The surrogate documents were then treated as a single collection and indexed in the SPIDER system, allowing us to construct German-French and French-German similarity thesauri as described in [Sheridan et al., 1997]. For cross-language retrieval in each direction the full topic description was expanded from the topic language to the target language terms most similar to the topic concept using the similarity thesaurus. The target language terms from the similarity thesaurus were then submitted as a query, followed by an automatic pseudo relevance feedback loop using the top 10 initially ranked documents to expand the query by the top 50 features. Our runs ETHfd1 and ETHdf1 use the 25 most similar terms from the similarity thesaurus as the query pseudo-translation and runs ETHfd2 and ETHdf2 take the 50 most similar terms.

4.2 English/German Retrieval

We decided that the Associated Press documents of the English collection were not so likely to be comparable to the SDA documents in German, which therefore ruled out a similar approach to building similarity thesauri as that used for the German-French experiments. We were however able to make use of the resource provided by the University of Maryland and the LOGOS Corporation, who translated the documents of the SDA German collection into English. This presented us with a bilingual parallel corpus on which to construct similarity thesauri. For our experiments however, we decided to take the stance which claims that full translations of all documents is not a practical approach in general for cross-language retrieval. We therefore did not rely on the complete document translations in any of our experiments. Instead, we compromised by extracting from each document and each translation the title and lead fields, representing a summary of abstract of each document. While assuming that full document translation for large collections is impractical, we verified that translation of titles and abstracts was a viable alternative by also automatically translating the titles and leads of the German NZZ collection using an off-the-shelf PC-based system called T1. Our German-English similarity thesauri for both German-to-English and English-to-German retrieval were based on the surrogate bilingual document collection constructed out of the German *document summaries* of the SDA and NZZ (except for the month of March which caused translation problems) collections together with their MT-translated equivalents (LOGOS for SDA and T1 for NZZ).

For our English-to-German retrieval experiments, each run included as a first step the expansion of the source-language topic using a retrieval and pseudo relevance feedback loop over the English document collection, using the top ten documents as relevant and expanding by the top 50 features. The expanded query was then submitted to whatever query translation method was being

used in the particular run. This pre-translation query expansion was only used for English-to-German as a post-experiment analysis showed that the query expansion tended to broaden the query topic too much, whereas we expected that more specific query topics would do better through translation (especially since similarity thesauri introduce a further query expansion and broadening effect). We therefore expect our English-to-German runs which used the pre-translation expansion to underperform the similarity-configured equivalent runs in other language pairs which did not use pre-translation expansion.

Submitted runs denoted by ETHed2 and ETHed3 used the similarity thesaurus trained over the document summaries and translations. ETHed2 consisted of pre-translation pseudo relevance feedback query expansion to 50 features on the AP collection, followed by pseudo-translation to the 25 most similar German terms which were then submitted as a German query, followed by a pseudo relevance feedback loop in German, assuming the top 10 documents relevant and expanding by the top 50 features. The same process held for ETHed3, except the similarity thesaurus returned the most similar 50 German terms. One observation worth making here is that the cross-language runs based on similarity thesaurus constructed using MT produced parallel corpora are still susceptible to errors introduced during the MT process. This is illustrated by the fact that a lookup of the similarity thesaurus for English features similar to the German *Waldheim* will return *forest* as a similar features because *Waldheim* has been consistently translated as *"forest home"* in the underlying corpus. Building similarity thesauri over artificial corpora is therefore less reliable than using manually created corpora.

Our run ETHed1 was used to explore if machine translation of documents summaries was enough in itself to achieve acceptable cross-language performance. The English topic was expanded using pseudo relevance feedback over the AP collection as described above and the resulting query was submitted to the *English* translations of the German SDA and NZZ documents. The ranked list of returned translations (whose DOCNO id still identifies the original German document) was submitted as the result. On the other hand, run ETHed4 explored the usefulness of machine translation for straight query translation. Query topics were directly translated from English to German using the off-the-shelf T1 MT system and the German translations submitted, followed by a pseudo relevance feedback loop in German, assuming the top 10 documents relevant and expanding by the top 50 features. Note that run ETHed4 did *not* use pre-translation query expansion.

Our experiments in German-to-English retrieval were in many ways similar to the English-to-German runs described above. Since the English documents were not translated to German however, we could not attempt an equivalent of run ETHed1 which submitted the queries against translated versions of document summaries. We also dropped the pre-translation query expansion step having decided that it seemed likely to lead to a loss of effectiveness.

Runs denoted ETHde1 and ETHde2 used the similarity thesaurus trained

over the document summaries and translations. ETHde1 consisted of pseudo-translation of the German topic to the 25 most similar English terms which were then submitted as a query, followed by a pseudo relevance feedback loop, assuming the top 10 documents relevant and expanding by the top 50 features. The same process held for ETHde2, except the similarity thesaurus returned the most similar 50 German terms. ETHde3 was the mirror of run ETHed4, directly translating the German topic to English using the T1 MT system and submitting the translation as query, followed by pseudo relevance feedback.

5 Spoken Document Retrieval

The Spoken Document Retrieval (SDR) track provided the perfect opportunity for evaluating our probabilistic weighting approach to indexing and retrieval of audio recordings [Wechsler and Schäuble, 1995]. Unfortunately our previous work had focused on German recordings, so the first step in our SDR participation was the development of an equivalent speech recognition system for English. We therefore built a speaker-independent phoneme recogniser for English speech using the HTK Toolkit [Young et al., 1993]. In comparison with the transcriptions provided as part of the track, our system achieved a phoneme recognition rate of 62.47%. For translating textual queries, reference transcriptions, and IBM's word-based recognition output to phonemic transcriptions we use an adapted version of the Carnegie Mellon Pronouncing Dictionary [CMU, 1995] with an additional rule-based translation program [Wasser, 1985] for translating out-of-vocabulary words.

Our retrieval method for spoken documents consists of four steps: slot detection, probability estimation, weighting and document ranking. The slot detection module locates possible occurrences of a query word in a spoken document. It also returns partial matches in order to cope with phoneme recognition errors. A similar method was employed in [Mittendorf et al., 1995] for retrieval of OCR-corrupted documents. For each identified slot we then determine the probability that the query word was actually spoken in that slot. The estimation employs a phoneme string similarity function enhanced with a phoneme confusion matrix which we derived from the SDR training set. Unlikely slot occurrences are then pruned using a threshold function. The aim is to filter out slots with lower probabilities as these are often "false alarms". The threshold is also sensitive to the length of query features, therefore favouring longer features which then to provide better clues of document relevance.

Estimated slot probabilities then contribute to the computation of an *expected feature frequency* for features in documents [Schäuble, 1997]. We adjust the computed expected feature frequencies however, compensating for systematic errors observed under training conditions. This is a similar compensation as was applied in the TREC-5 confusion track for OCR data [Ballerini et al., 1996]. Given values for expected feature frequencies and inverse expected collection fre-

quencies, we then turned our attention to the retrieval function. In particular, we examined the issue of document length normalisation and developed a new document weighting scheme that uses what we call *logarithmic document length normalisation.* This scheme gave better results than the standard methods (cosine, pivoted document length normalisation etc.) when applied on the training collection. We also introduced a *logarithmic feature length normalisation* with the aim of adding weight to longer than average indexing features in a given query. This follows our belief that longer query features have more information value and, furthermore, are more likely to be detected in spoken documents. The logarithmic feature length normalisation alone was seen to accomplish an 18% improvement in average precision when performing known-item searches on the training collection with 13 queries (the SDR training queries plus some we devised ourselves). These compensation and normalisation methods were included in our supplementary submissions, ETHS2 based on our own speech recognition and ETHB2 based on IBM output.

For our obligatory runs ETHS1, ETHB1 and ETHR1 we added a compensation of documents weights to smooth the differences between the number of slots in a given document for a particular query feature compared to the average number of slots for that feature over the whole document collection. This compensation may be called *pivoted number-of-slots normalisation* since it resembles the pivoted document length normalisation of [Singhal et al., 1996]. A further new compensation measure addresses the variability in the ratio of Expected RSV to actual RSV for a given document depending on the number of query features in that document. This variability can lead to corruption of the ranked document list. We can prove theoretically the existence of such a variability and have confirmed it in our experimental experience. Using an analysis of the training data to tune the parameters of our compensation and normalise the variability in the ERSV/RSV ratio we have found that this correction delivers a 19% improvement in average precision on our training collection.

Note that in respect to the estimation of slot probabilities, our submitted runs on the reference data (perfect textual documents) and baseline data (word-based recognition) used slots which had been positively identified (probability 1) since in these cases we only want exact feature matching. We used the full probabilistic slot matching for the runs submitted based on our own phoneme-level speech recogniser (ETHS1, ETHS2).

Although we are aware that our own speech recognition system needs much further improvement, we consider the probabilistic retrieval approach described above as the major contribution of our work in this track. Initial investigations of the performance of this probabilistic approach compared to the SPIDER system used in the TREC-5 adhoc retrieval task show no significant difference in performance. We therefore aim to achieve comparable performance on spoken document retrieval tasks using current state-of-the-art speech recognition systems as we achieve on perfect text documents with the SPIDER system.

6 Conclusion

We have described here our participation in the TREC-6 evaluation with submission in the main routing task and in the tracks concerned with Chinese text retrieval, cross-language information retrieval, and spoken document retrieval. The main themes of this work can be summarised as:

Routing - We have improved our U-measure for feature selection by including features from query titles. We group semantically related features on a per query basis and generate queries of multiple semantic groups for each topic. The results of the feature group retrieval are combined, also on a per query basis, with results of retrieval using the Lnu.ltn retrieval method and a retrieval method using feature co-occurrence matrices. The combination of ranked lists generated by different methods has been shown to lead to improvements in performance when compared to the performance of the individual methods.

Chinese - We have built upon our earlier efforts by including a new manually composed stoplist which has helped to reduce our index space by 45% without adversely affecting in retrieval performance. Chinese text is indexed using character-bigrams, although English words and numeric strings are identified and indexed whole.

Cross-Language - The SPIDER system indexing engine includes stoplists and stemming or word normalisation modules for each of the languages covered. We have further evaluated our similarity thesaurus approach for cross-language retrieval, constructing similarity thesauri over comparable corpora when available and investigating the use of parallel collections created by applying machine translation to documents or document abstracts when no comparable corpus is available. We have also tested the effectiveness of retrieval directly using machine translation output, both with document abstract translations or query translations.

Speech Retrieval - Apart from developing our own speech recognition system for English, we have made major innovations in our probabilistic approach to retrieval from errorful information. These include new document length and feature length normalisations in the weighting stage and a new compensation measure in the retrieval method which normalises the variability in Expected RSV compared to the actual RSV which arises across documents which contain different numbers of query features.

Our work on each of these themes is ongoing and we hope that several aspects of the work presented here can be further developed and reported much more formally in the information retrieval literature.

References

[CMU, 1995] (1995). *Carnegie Mellon Pronouncing Dictionary (cmudict.0.4, 1995).* http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[Baayen et al., 1993] Baayen, R., Piepenbrock, R., and van Rijn, H. (1993). The CELEX Lexical Database. Technical report, Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

[Ballerini et al., 1996] Ballerini, J., Büchel, M., Domenig, R., Knaus, D., Mateev, B., Mittendorf, E., Schäuble, P., Sheridan, P., and Wechsler, M. (1996). SPIDER Retrieval System at TREC5. In *TREC-5 Proceedings.*

[Mittendorf et al., 1995] Mittendorf, E., Schäuble, P., and Sheridan, P. (1995). Applying Probabilistic Term Weighting to OCR Text in the case of a Large Alphabetic Library Catalogue. In *Proceedings of the 18th ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA*, pages 328–335.

[Porter, 1980] Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3):130–137.

[Qiu, 1995] Qiu, Y. (1995). *Automatic Query Expansion Based on a Similarity Thesaurus.* PhD thesis, Swiss Federal Institute of Technology.

[Schäuble, 1997] Schäuble, P. (1997). *Multimedia Information Retrieval—Content-Based Information Retrieval from Large Text and Audio Databases.* Kluwer Academic Publishers, Boston/London/Dordrecht.

[Sheridan and Ballerini, 1996] Sheridan, P. and Ballerini, J. P. (1996). Experiments in Multilingual Information Retrieval using the SPIDER System. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland*, pages 58–65.

[Sheridan et al., 1997] Sheridan, P., Braschler, M., and Schäuble, P. (1997). Cross-Language Information Retrieval in a Multilingual Legal Domain. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, Pisa, Italy*, pages 253–268.

[Singhal et al., 1996] Singhal, A., Buckley, C., and Mitra, M. (1996). Pivoted Document Length Normalization . In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland*, pages 21–29.

[Wasser, 1985] Wasser, J. A. (1985). *English to Phoneme Translation.* Program in public domain, ftp://ftp.doc.ic.ac.uk/packages/unix-c/utils/phoneme.c.gz.

[Wechsler and Schäuble, 1995] Wechsler, M. and Schäuble, P. (1995). Speech retrieval based on automatic indexing. In Ruthven, I., editor, *Proceedings of the Final Workshop on Multimedia Information Retrieval (MIRO'95)*, Electronic Workshops in Computing, Glasgow. Springer.

[Wechsler et al., 1997] Wechsler, M., Sheridan, P., and Schäuble, P. (1997). Multi-language text indexing for internet retrieval. In *Proceedings of the 5th RIAO Conference, Computer-Assisted Information Searching on the Internet, Montreal, Canada*, pages 217–232.

[Young et al., 1993] Young, S., Woodland, P., and Byrne, W. (1993). *HTK Version 1.5: User, Reference & Programmer Manual.* Entropic Cambridge Research Laboratory, Sheraton House, Castle Park, Cambridge CB3 OAX, England.

[Zavrel and Veenstra, 1995] Zavrel, J. and Veenstra, J. (1995). The Language Environment and Syntactic Word-Class Acquisition. In *Proceedings of the Groningen Assembly on Language Acquisition (GALA95)*, pages 365–374.

Phrase Discovery for English and Cross-language Retrieval at TREC-6

Fredric C. Gey and Aitao Chen
UC Data Archive & Technical Assistance (UC DATA)
gey@ucdata.berkeley.edu
aitao@sims.berkeley.edu
University of California at Berkeley, CA 94720

January 20, 1998

Abstract

Berkeley's experiments in TREC-6 center around phrase discovery in topics and documents. The technique of ranking bigram term pairs by their expected mutual information value was utilized for English phrase discovery as well as Chinese segmentation. This differentiates our phrase-finding method from the mechanistic one of using all bigrams which appear at least 25 times in the collection. Phrase finding presents an interesting interaction with stop words and stop word processing. English phrase discovery proved very important in a dictionary-based English to German cross language run. Our participation in the filtering track was marked with an interesting strictly Boolean retrieval as well as some experimentation with maximum utility thresholds on probabilistically ranked retrieval.

1 Introduction

Berkeley's participation in the TREC conferences has provided a venue for experimental verification of the utility of algorithms for probabilistic document retrieval. Probabilistic document retrieval attempts to place the ranking of documents in response to a user's information need (generally expressed as a textual description in natural language) on a sound theoretical basis. The approach is, fundamentally, to apply Bayesian inference to develop predictive equations for probability relevance where training data is available from past queries and document collections. Berkeley's particular approach has been to use the technique of logistic regression. Logistic regression has by now become a standard technique in the discipline of epidemiology for discovering the degree to which causal factors result in disease incidence [8, Hosmer and Lemeshow,89]. In document retrieval the problem is turned around, and one wishes to predict the incidence of a rare disease called 'relevance' given the evidence of occurrence of query words and their statistical attributes in documents.

In TREC-2 [3] Berkeley introduced a formula for ad-hoc retrieval which has produced consistently good retrieval results in TREC-2 and subsequent TREC conferences TREC-4 and TREC-5.

The logodds of relevance of document D to query Q is given by

$$\log O(R|D,Q) = -3.51 + \frac{1}{\sqrt{N}+1}\Phi + 0.0929 * N \qquad (1)$$

$$\Phi = 37.4 \sum_{i=1}^{N} \frac{qtf_i}{ql + 35} + 0.330 \sum_{i=1}^{N} \log \frac{dtf_i}{dl + 80} - 0.1937 \sum_{i=1}^{N} \log \frac{ctf_i}{cf} \quad (2)$$

where

N is the number of terms common to both query and document,
qtf_i is the occurrence frequency within a query of the ith match term,
dtf_i is the occurrence frequency within a document of the ith match term,
ctf_i is the occurrence frequency in a collection of the ith match term,
ql is query length (number of terms in a query),
dl is document length (number of terms in a document), and
cf is collection length, i.e. the number of occurrences of all terms in a test collection.

The summation in equation (2) is carried out over all the terms common to query and document.

This formula has also been used, with equal success, in document retrieval with Chinese and Spanish queries and document collections of the past few TREC conferences. We utilized this identical formula for German queries against German documents in the cross-language track for TREC-6.

Berkeley's approach, in the past, has been to concentrate on fundamental algorithms and not attempt refinements such as phrase discovery or passage retrieval. However in doing further research in the area of Chinese text segmentation [2] we applied a technique from computational linguistics which seemed to show promise for rigorous discovery of phrases from statistical evidence based upon word frequency and word co-occurrence in document collections. Thus for TREC-6 we have begun the investigation of how to obtain and use phrases within the context of probabilistic document retrieval.

2 Phrase discovery using expected mutual information

The usual method at TREC (by many other groups) for choosing phrases has been to mechanistically choose all two word combinations which occur more than 'n' times in the collection (where n=25 has been the usual threshold). Other groups have used natural language processing techniques (rule and dictionary-based) to parse noun phrases. Berkeley's approach for TREC-6 was compute the mutual information measure between word combinations using individual and word co-occurrence frequency statistics:

$$MI(t_1, t_2) = \log_2 \frac{P(t_1, t_2)}{P(t_1)P(t_2)}$$

High values of this measure indicate a positive association between words. Near zero values indicate probabilistic independence between words. Values less than zero indicate a negative correlation between words (i.e. if one word occurs, the other word is not likely to occur next to it). Our experiments indicated that values of MI greater than 10 almost always identified proper nouns such as (for TREC topic 001 in routing) 'Ivan Boeski' and 'Michael Milkin'. This technique identifies important phrases such as 'unfriendly merger' which occur only 5 times in the collection. Berkeley used a cutoff of MI = 3.00. However, when both of the component words are commonly occurring words, the expected mutual information value will be a small value. In this case the mutual information technique may fail to identify high frequency phrases (such as 'educational standard' with MI = 1.70 which occurs 399 times in the 5 TREC disks).

Phrase discovery has an important interaction with stopword processing. For TREC-6 adhoc topic 340, the title query 'Land Mine Ban' processes to 'land' and 'ban' because 'mine' is a

stopword. Interestingly this does not affect the Description Field for that topic which contains the phrase 'land mines' which stems to 'land mine'. Berkeley chose to identify phrases before stopword processing. This produces other interesting phrases such as 'for example' and 'e g', although they may not be particularly discriminating. Because we made this processing decision after examining the parsing of the title for topic 340, we did not submit a short title run for TREC-6. We do, however, include a short title result below for comparison purposes.

Another important question is whether to retain the individual word components of phrases or to remove them. Our experiments indicate that performance deteriorates upon removal of individual word components of phrases, at least for ad-hoc retrieval.

3 Ad-hoc Experiments

Berkeley's ad-hoc runs for TREC-6 utilized the new phrase discovery method as well as a new formula to incorporate phrases into probabilistic training. Our decision to modify the TREC-2 formula was based upon the observation that phrases have a very different pattern of occurrence in the collections than individual terms. The principle thrust of the change was to separate out a component which utilized the statistical clues for phrases as distinct from one which used single term statistical attributes. After training using logistic regression on relevance judgments for disks 1-4, the formula was as follows:

The logodds of relevance of document D to query Q is given by

$$\log O(R|D,Q) = -3.9912 + \frac{1}{\sqrt{N_t}+1}\Phi_t + 0.1281 * N_t + \frac{1}{\sqrt{N_p}+1}\Phi_p - 0.3161 * N_p \tag{3}$$

$$\Phi_t = 36.5904 \sum_{i=1}^{N_t} \frac{qtf_i}{ql_t+35} + 0.3938 \sum_{i=1}^{N_t} \log \frac{dtf_i}{dl_t+80} - 0.2147 \sum_{i=1}^{N_t} \log \frac{ctf_i}{cf_t} \tag{4}$$

$$\Phi_p = 6.5743 \sum_{i=1}^{N_p} \frac{qpf_i}{ql_p+10} + 0.0959 \sum_{i=1}^{N_p} \log \frac{dpf_i}{dl_p+25} - 0.1182 \sum_{i=1}^{N_p} \log \frac{cpf_i}{cf_p} \tag{5}$$

where

N is the number of terms common to both query and document,
qtf_i is the occurrence frequency within a query of the ith match term,
dtf_i is the occurrence frequency within a document of the ith match term ,
ctf_i is the occurrence frequency in a collection of the ith match term, ql_t is query length (number of single terms in a query),
dl_t is document length (number of single terms in a document), and
cf_t is collection length, i.e. the number of occurrences of all single terms in a test collection.
qpf_i is the occurrence frequency within a query of the ith match phrase,
dpf_i is the occurrence frequency within a document of the ith match phrase ,
cpf_i is the occurrence frequency in a collection of the ith match phrase, ql_p is query length (number of phrases in a query),
dl_p is document length (number of phrases in a document), and
cf_p is collection length, i.e. the number of occurrences of all phrases in a test collection.

Run	Brkly21	Brkly22	Brkly23	Title	Words
Formula	TREC-6	TREC-6	TREC-2	TREC-2	TREC-2
Query	Description	Long	Manual	Title	Long
Phrase	Yes	Yes	Yes	Yes	No
Expansion	Yes	Yes	No	No	No
Overall	0.1376	0.2021	0.2282	0.2102	0.2054
0.00	0.5668	0.7105	0.6558	0.6001	0.7191
0.10	0.3298	0.4449	0.4885	0.4442	0.4416
0.20	0.2333	0.3456	0.3745	0.3337	0.3505
0.30	0.1802	0.2704	0.3128	0.2753	0.2747
0.40	0.1399	0.2218	0.2623	0.2425	0.2164
0.50	0.1160	0.1903	0.2182	0.2025	0.1721
0.60	0.0963	0.1398	0.1677	0.1619	0.1365
0.70	0.0611	0.1029	0.1193	0.1234	0.1070
0.80	0.0209	0.0472	0.0604	0.0843	0.0704
0.90	0.0070	0.0226	0.0221	0.0526	0.0195
1.00	0.0030	0.0119	0.0126	0.0475	0.0089
relevant	1615	2547	2583	2321	2382
5 docs	0.3680	0.4680	0.4880	0.3720	0.4400
10 docs	0.2940	0.4080	0.4320	0.3500	0.3880
15 docs	0.2680	0.3693	0.3813	0.3227	0.3520
20 docs	0.2450	0.3440	0.3510	0.3050	0.3340
30 docs	0.2160	0.3053	0.3140	0.2767	0.2960
100 docs	0.1286	0.1932	0.2112	0.1912	0.2000
200 docs	0.0878	0.1365	0.1433	0.1306	0.1385
500 docs	0.0502	0.0800	0.0816	0.0762	0.0786
1000 docs	0.0323	0.0509	0.0517	0.0464	0.0476
R-Precision	0.1675	0.2422	0.2612	0.2419	0.2500

Table 1: TREC-6 Adhoc Results

The summation in equations (4) and (5) is carried out over all the terms or phrases common between query and document.

The size of the training matrix produced was 3,812,933 observations. The normalization by collection length (single terms and phrases) was done by counting total occurrences of all single terms/pairs in the collection. These are:

158,042,364 single terms
34,018,769 pairs

Our official runs were Brkly21 (long topic run) Brkly22 (description field run) and Brkly23 (manual query reformulation). As can be seen from the table the description field run was significantly below the long topic run, continuing a pattern begun in TREC-5. Our unofficial run on the title field produced almost equivalent performance to the long field, attributable to the precision by which titles capture the essential meaning of the topics. We also ran a long query run using only the TREC-2 formula, and were dismayed to find that the phrase formula failed to improve upon single terms. It seems that phrases, which offer significantly more precise capture of topic meaning, have yet to be exploited properly by our probabilistic training.

4 Routing Experiments

Berkeley's routing runs for TREC-6 follow on the spirit of our routing runs of TREC-5. In all routing methodology the key problem is to choose additional terms to add to each query based upon documents found to be relevant in previous TREC runs. Several measures have been proposed to choose such terms, including the χ^2 measure which Berkeley used in TREC-3 and TREC-4. This measure ranks terms by the degree to which they are dependent upon relevance. In earlier TRECs, Berkeley did massive query expansion by choosing all terms associated with relevance at the 5 percent significance level. In TREC-5 this resulted in a variable number of terms per query from a minimum of 714 to a maximum of 3839 with a mean of 2032 terms over the 50 queries.

In TREC-5 Berkeley introduced the idea of using logistic on the term frequency in documents for the 15 most important terms in the ranking. This produced an approximately 20 percent improvement over the massive query expansion. Further investigations following TREC-5 showed equivalent performance improvements for the top 3 and 5 terms as well, [5] and that adding more terms achieved higher precision at the expense of total documents retrieved in the top 1000 documents.

As can be imagined, processing for 100,000 query terms over 50 documents becomes an i/o and cpu intensive task. Moreover, when we began a similar χ^2 selection for the 43 old queries of TREC-6, it produced 486,308 query terms, or 11,309 per query. The processing task for such queries seemed insurmountable for our limited resources. Thus we took to choosing a χ^2 cutoff at the 0.001 significance level.

At the same time we began investigating the U-Measure used by ETH in TREC-5 [4] also known as the Correlation Coefficient used in a text categorization study by Ng and others [9]. This measure is claimed to improved upon χ^2 by eliminating negative correlations between terms and relevance. Indeed our initial experiments showed that choice of the top 50 terms by u-measure ranking would produce results close to massive query expansion using χ^2. This was thus the method by which we choose terms for addition to the query, after retrieving all terms which satisfied a significance cutoff of 0.001 for the U-measure. We also performed logistic regression training on the term frequency in document for the top 5 and top 15 terms. These became our official runs BRKLY19 and BRKLY20.

Unfortunately the uniform application of a 0.001 significance level adversely affected the new routing topics 10001-10004 for which there was limited training data. Thus our choice of cutoff produced less than 28 additional terms for each of these queries, including these ten terms for topic 10003 (Privatization in Peru) – 'span-feb', 'span', 'priv', 'editor-report', 'cop', 'editor', 'roundup', 'feb', 'through-febru', 'la' – hardly very discriminating terms. It is not surprising that our performance on this query was among the worst of our performances when compared to the median. Choice of a 5 percent significance level would surely have produced better queries.

Another problem which we immediately encountered in processing the routing data was massive document duplication in the initial files of FBIS2. For example a simple pattern search of headers H3 reveals over 50 copies of the document headed by

```
<H3> <TI>    Thomson-CSF, Thorn EMI Defense Link-Up </TI></H3>
```

Fortunately this massive duplication seems to be confined to the first 20 files of the collection, although a random selection of other files revealed a few duplicates. As far as results are concerned, we have not spent time examining for duplicate documents, but we have determined that our top ranked two documents

```
003 Q0 FB6-F144-0008        1   1.000000 Brkly20
003 Q0 FB6-F144-0025        2   1.000000 Brkly20
```

for the Brkly20 run for query 003 (Japanese joint ventures) are identical documents with different document ids.

5 Tracks

For TREC-6 Berkeley participated in the Filtering, Chinese, and Cross-language tracks. An independent effort was mounted for the interactive track which is summarized in a separate paper. Berkeley had participated in the Chinese track in TREC-5 but this was our first participation in the Filtering track. For Cross-language, Berkeley submitted runs for English queries against German documents.

5.1 Cross-language: English queries against German documents

Berkeley decided to participate in the cross-language track in order to once again test the robustness of our probabilistic algorithm for ad-hoc document retrieval which has performed so well for Chinese and Spanish retrieval [6]. Our German-German run used the TREC-2 algorithm unchanged from its English implementation. For both our German-German and English-German runs we recognized the importance of phrase discovery which Ballesteros and Croft [1] have found to be paramount in effective cross-language retrieval. In English to German this becomes paramount because of the propensity for German to form compounds of single words equivalent to phrases in English. For example, the phrase 'air pollution' of topic CL6 can become the word 'Luftverschmutzung' in German, whereas the words 'air' and 'pollution' submitted separately to a dictionary do not provide the same meaning. The choice in dictionary retrieval is between obtaining only individual words which have little relationship to the phrase or obtaining all possible compound variations of the particular individual words. The former course results in missing the particular compound, while the latter results in obtaining a large set of noise words.

Initially we were unable to obtain an English-German dictionary and discovered a WWW dictionary (http://www.bg.bib.de/ã2h6bu) We had to write a cgi script which submitted English words and phrases and captured the output of the German translation. Since the transmission was subject to timeout failures, several runs had to be pooled and duplicate entries removed to obtain a final query.

Unlike our processing of the main track documents and queries, we did not retain the individual word components of discovered phrases. Finally, when English words were not found in the dictionary we kept the English word in the German query under the assumption that proper names (Kurt Waldheim is a good example) would be the same in both languages. These principles guided our English to German automatic run BrklyE2GA.

Our manual run BrklyE2GM was produced by the same processing guidelines except that the English source was manually modified in much the same way as our main track manual modification. Phrases such as 'a relevant document will discuss' were removed (query reduction) while queries were also expanded to include reasonable specifics. In particular, topic CL13 on the Middle East peace process, specific country and place names such 'Israel', 'Egypt', 'Syria', 'west bank', 'golan heights' were added to the query. Unfortunately the dictionaries used did not contain translations for all geographic names so the value of the enhancement is unclear.

Our results are as follows: our German-German run (BKYG2GA) achieved average precision of .2845 over 21 judged topics (versus 0.3417 over the 13 topics judged before the conference), while our English-German automatic run had average precision of 0.1305 and the English-German manual run had average precision of 0.1822. Interestingly for topic CL24 on 'teddy bears', the precision

	BKYG2GA	BKYE2GM	BKYE2GA	XTGBL	XTETH
total rel	992	992	992	992	992
rel ret	675	422	295	452	425
avg prec	0.2845	0.1822	0.1305	0.1185	0.1831
0.00	0.6894	0.4647	0.3766	0.3789	0.4620
0.10	0.5259	0.3350	0.2390	0.2589	0.3377
0.20	0.4809	0.2871	0.1778	0.2035	0.2761
0.30	0.4402	0.2462	0.1598	0.1630	0.2321
0.40	0.3614	0.2237	0.1429	0.1279	0.2095
0.50	0.2933	0.1726	0.1337	0.1118	0.1706
0.60	0.2391	0.1356	0.1186	0.0886	0.1557
0.70	0.1556	0.1058	0.0977	0.0601	0.1277
0.80	0.0810	0.0812	0.0724	0.0460	0.0903
0.90	0.0269	0.0649	0.0274	0.0354	0.0598
1.00	0.0088	0.0059	0.0048	0.0263	0.0318

Table 2: TREC-6 Cross-Language Retrieval Results

of 0.8330 for our manual run exceeded the best precision of 0.7541 for the 10 German-German monolingual runs. This can be directly attributed to the process of query reduction.

On the other hand, the manual query for topic CL2 (marriages and marriage customs) had a disastrous reduction in precision from 0.1524 (BKYE2GA) to 0.0492 (BKYE2GM), which may be attributable to the addition of the word 'customs' (as in marriage customs) which produced numerous translations.

One question is to the degree of overlap between monolingual and crosslingual retrieval. We analyzed the overlap between our German-German and English-German automatic runs and found that 14,894 documents in common among the 25000 documents retrieved by each run. We did not examine the overlap in the top 50 documents.

Since the conference we purchased the GlobalLink web translation package and used it to translate the topics from English to German. This automatic run (XTGBL) produced a precision of 0.1185, worse than our dictionary based automatic run, while at the same time retrieving more relevant documents (452) than any other cross-language run. Paraic Sheridan of the ETH group kindly supplied their machine translation of the English topics which used the T1 text translator which incorporates the Langenscheidt Dictionary. This run (XTETH) achieved a precision of 0.1831, slightly better that Berkeley's manual run.

Table 2 provides a detailed comparison of all our experiments.

5.2 Filtering

TREC-6 was the Berkeley group's first participation in the filtering track. While our entry is a straightforward probabilistic ranking with threshold approach, some interesting twists appeared as we began to work on the problem. First, we used an approach to query development identical to our TREC-6 routing approach (basically query expansion using statistical measures of Chi Square and U-measure, as well as logodds of relevance) trained only on the FBIS disk5 training set. For some topics, important query terms proved to be identical to those for routing training, while for other queries a dramatically different set of terms emerged. In addition, we use logistic regression on the term frequencies of the 5 most important terms. Because of the paucity of training data for some queries, the regression would not converge for four of the 47 filtering topics, so we had to use

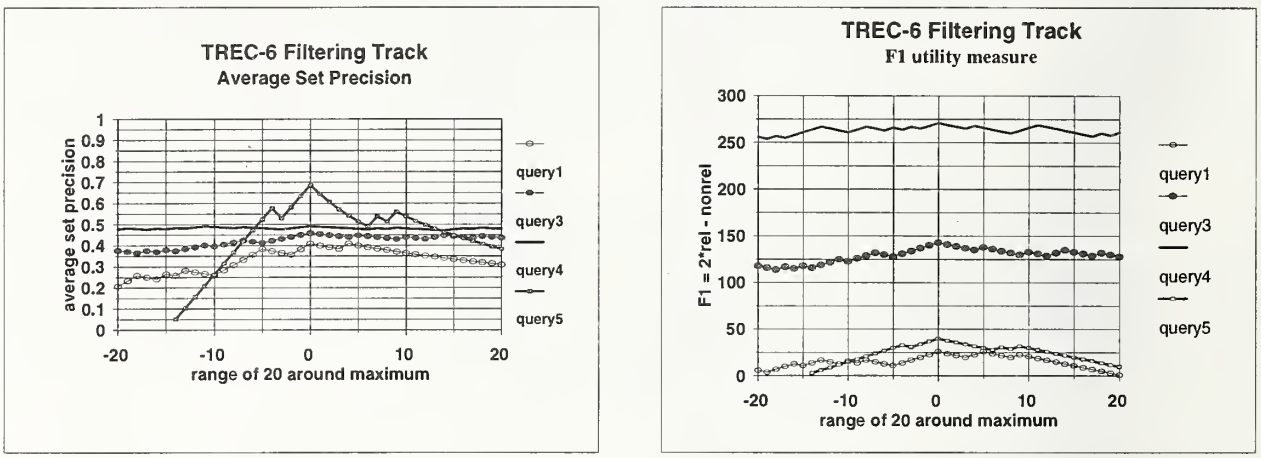


Figure 1: Filtering thresholds for ASP and F1.

a completely different thresholding mechanism for those four topics.

Our probability threshold was chosen for each utility measure based upon maximizing the utility over the training data. However examination of the distribution of utilities around the maximum showed quite different behavior patterns for different topics-some maxima were quite crisp while others were fuzzy or uncertain. Furthermore, for crisp thresholds (ones where the maximum utility is significantly higher than the surrounding utilities), it is unclear whether to choose that threshold or to lower the threshold in the direction of the next-highest values.

Figure 1 plots the values of average set precision for 20 document ranks on either side of the maximum value for the first four trec queries. As can be seen the maximum is crisp only for TREC query 005. This query is also the only one where the maximum is achieved before 20 documents have been ranked. On the other hand query 001 has a very fuzzy threshold, achieving close to the maximum at document ranks well beyond the actual maximum. It is unclear what value should have been used for thresholding for this query. The choice of thresholds from ranked retrieval appears to be a fundamental research problem.

Finally Berkeley decided to submit a pure Boolean run which consisted of those documents which contained all 5 most important query terms for each topic. We submitted this run (BKYT6BOOL) to be evaluated by all three evaluation measures. The number of documents retrieved by this method was dramatically different from the probability threshold results. By all measures (when averaged over 47 queries) the Boolean retrieval performed much worse than probabilistic retrieval with thresholding. Interestingly enough, however, the retrieval of 52 documents for topic 001 scored the maximum for all three performance measures. For that topic the five terms used for

coordination retrieval were 'commit' 'fair trad' 'trad' 'fair' 'ftc'.

5.3 Chinese

Because Chinese text is delivered without word boundaries, automatic segmentation of text into imputed word components is a prerequisite to retrieval. One group of word segmentation methods are based on dictionary. Berkeley believes that the coverage of the dictionary over the collection to index can have significant impact on the retrieval effectiveness of a Chinese text retrieval system that uses a dictionary to segment text. In TREC-5 [7], we combined a dictionary found on the web and entries consisting of words and phrases extracted from the TREC-5 Chinese collections to create a dictionary of about 140,000 entries and we used the dictionary to segment the Chinese collection. This dictionary certainly is not small in size, yet we found that the dictionary did not include many of the proper names such as personal names, transliterated foreign names, company names, university and college names, research institutions and so on. Our focus in Chinese track of TREC-6 was on automatic and semi-automatic augmentation of the Chinese dictionary which we used to segment the Chinese collection.

Based on the observations that personal names are often preceded by title names and followed by a small group of verbs such as *say*, *visit*, *suggest* et al, and the first name, middle name and the last name of a transliterated foreign name are separated by a special punctuation mark, we constructed a set of pattern rules by hand to extract any sequence of characters in the text that matches any pattern rule. We then went through the list by hand to remove the entries that are not personal names.

In Chinese text, the items (such as names) in a list are uniquely marked by a special punctuation mark. We wrote a simple program to take out any sequence of characters flanked by the special punctuation mark. The technique seems to be quite productive for it produced over 10,000 entries from the TREC-5 Chinese collection. There are, of course, some entries that are not meaningful. The appendix contains a sample text excerpt and the names (country names and company names) that were extracted from the excerpt.

Berkeley submitted two runs, named BrklyCH3 and BrklyCH4 respectively, for the Chinese track. BrklyCH3 is the run using the original long queries with automatic query expansion and BrklyCH4 is the run based on the manually reformulated queries. For both runs, the collection was segmented using the dictionary-based maximum matching method. For BrklyCH3, an initial retrieval run was carried out to produce a ranked list of documents, then 20 new terms were selected from the top 10 ranked documents for each query. The selected terms are those that occur most frequently in the top 10 documents in the initial ranked list. The chosen terms were added to the original long queries to form the expanded queries. A final run was carried out using the automatically expanded queries to produce the results in BrklyCH3. For both runs, the documents were ranked by the probability of relevance estimated using the Berkeley's TREC-2 adhoc retrieval formula. For BrklyCH4, we spent about 40 minutes per query to manually reformulate each query by 1) removing non-content words from the original queries; 2) adding new words found in the collection to the original queries; and 3) adjusting the weights assigned to each term in the queries.

6 Conclusions and Acknowledgments

In our TREC-6 experiments for the main tasks and tracks, Berkeley worked primarily on extending our probabilistic document retrieval methods to incorporate two word phrases found using the ranking provided by expected mutual information measure. While these methods did not result in performance improvements for English retrieval, they were central in obtaining reasonable per-

formance in English queries against German documents in the crosslingual track. Our first foray into the Filtering task obtained reasonable results for precision by using threshold computations to truncate a ranked retrieval and obtain a pool of unranked documents. Clearly finding the proper threshold in transforming from ranked retrieval to document sets is a research problem which will require considerably more study.

We acknowledge the assistance of Jason Meggs who indexed and ran the German document collection and Lily Tam and Sophia Tang, computer science undergraduates who provided programming assistance and who helped in the manual reformulation of Chinese queries. This research was supported by the National Science Foundation under grant IRI-9630765 from the Database and Expert Systems program of the Computer and Information Science and Engineering Directorate.

References

[1] Lisa Ballesteros and W. Bruce Croft. Phrasal Translation and Query Expansion Techniques for Cross-Language Information Retrieval. In Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, editors, *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia*, pages 84–91, 1997.

[2] A. Chen, J. He, L. Xu, F. C. Gey, and J. Meggs. Chinese Text Retrieval Without Using a Dictionary. In A. Desai Narasimhalu Nicholas J. Belkin and Peter Willett, editors, *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia*, pages 42–49, 1997.

[3] W. S. Cooper, A. Chen, and F. C. Gey. Full Text Retrieval based on Probabilistic Equations with Coeffici ents fitted by Logistic Regression. In D. K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 57–66, March 1994.

[4] Ballerini et al. SPIDER Retrieval System at TREC-5. In D. K. Harman and Ellen Voorhees, editors, *The Fifth Text REtrieval Conference (TREC-5), NIST Special Publication 500-238*, pages 217–228, November 1997.

[5] F. C. Gey and A. Chen. Term importance in routing retrieval . In *Submitted for publication*, December 1997.

[6] F. C. Gey, A. Chen, J. He, L. Xu, and J. Meggs. Term importance, Boolean conjunct training, negative terms, and foreign language retrieval: probabilistic algorithms at TREC-5. In D. K. Harman and Ellen Voorhees, editors, *The Fifth Text REtrieval Conference (TREC-5), NIST Special Publication 500-238*, pages 181–190, November 1997.

[7] J. He, L. Xu, , A. Chen, J. Meggs, and F. C. Gey. Berkeley Chinese Information Retrieval at TREC-5: Technical Report. In D. K. Harman and Ellen Voorhees, editors, *The Fifth Text REtrieval Conference (TREC-5), NIST Special Publication 500-238*, pages 191–196, November 1996.

[8] David W. Hosmer and Stanley Lemeshow. *Applied Logistic Regression.* John Wiley & Sons, New York, 1989.

[9] H-T Ng, W-B Goh, and K-L Low. Feature Selection, Perceptron Learning, and a Useability Case Study for Text Categorization. In Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, editors, *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in In formation Retrieval, Philadelphia*, pages 67–73, 1997.

Appendix

An excerpt from a news article in the Xin Hua News collection.

——国外大公司投资活跃 去年年初，山东半岛各地就把吸引国外大公司、大商社投资做为重点，积极谋求与国际上资本雄厚、管理一流的大公司进行合作，先后有来自美国、德国、法国、瑞士、日本、韩国、香港的三十多个大公司前来投资办厂。

如美国道化学公司、戴诺润滑油公司、德国西门子公司、德固萨公司、瑞士汽巴嘉基公司、日本丰田公司、英国商业集团公司、韩国大宇公司、世运工业公司、香港中策公司、中银集团等。

Names extracted from the above paragraph include:

德国 German
法国 (France)
瑞士 (Switzerland)
日本 (Japan)
韩国 (German)
戴诺润滑油公司
德西门子公司
德固萨公司
瑞士汽巴嘉基公司
日本丰田公司
英国商业集团公司
韩国大宇公司
世运工业公司
香港中策公司

Cheshire II at TREC 6: Interactive Probabilistic Retrieval

Ray R. Larson Jerome McDonough

School of Information Management and Systems
University of California, Berkeley
Berkeley, CA 94720-4600
ray@sherlock.berkeley.edu

Abstract

This paper briefly describes the features of the Cheshire II system and how it was used in the TREC 6 Interactive track. The results of the interactive track are discussed and future improvements to the Cheshire II system are considered.

1 Introduction

The Cheshire II system was originally designed to apply probabilistic retrieval methods to searching in online library catalogs in order to help overcome the twin problems of topical searching that are pervasive in "second generation" Boolean online catalogs: search failure and information overload. It was originally intended to be a next-generation online catalog and full-text information retrieval system that would apply probabilistic retrieval methods to simple MARC records and clustered record surrogates (Classification clusters)(Larson 1991c; Larson, et al. 1996). Over time the system has been explanded to include support for full-text SGML documents (ranging from simple document types as used in the TREC database to complex full-text document encoded using the TEI and EAD DTDs) and support for full-text OCR from scanned page image files linked to SGML bibliographic records (as used in Berkeley's NSF/NASA/ARPA-sponsored Digital Library Project).

The Cheshire II system is currently being used in a working library environment (the UC Berkeley Mathematics, Statistics and Astronomy library) via a dedicated X window terminal and data on its use and acceptance by local library patrons and remote network users are being evaluated. The system is also being used as the primary text search engine for the UC Berkeley Environmental Digital Library project sponsored by NSF, NASA, and DARPA. It is also providing access to a number of diverse databases databases via the WWW using an HTTP to Z39.50 gateway.

The Cheshire II system includes the following features:

1. It supports SGML as the primary data base format of the underlying search engine, and provides support for full-text data linked to SGML metadata records. We support MARC format records for traditional online catalog databases using MARC to SGML conversion.

2. It is a client/server application where the interfaces (clients) communicate with the search engine (server) using the Z39.50 v.3 Information Retrieval Protocol. The system also provides a general Z39.50 Gateway with support for mapping Z39.50 queries to local Cheshire databases and to relational databases.

3. It includes a graphical direct manipulation interface (Sun SPARC) X terminals as well as a CGI interpreter version that combines client and server capabilities. These interfaces permit searches of the Cheshire II search engine as well as any other z39.50 compatible search engine on the network.

4. It allows users to enter queries as "free-text" (that is, normal English prose) statements of their interest or need. No formal "query language" or Boolean logic imposed on the user, although Boolean logic is available for those who desire it.

5. It uses probabilistic ranking techniques to match the user's initial query with documents in the database. In some databases it can provide two-stage searching where a set of "classification clusters"(Larson 1991c) for the database is first retrieved in decreasing order of probable relevance to the user's search statement. These can then be used to provide feedback about the primary topical areas of the query, and retrieve documents within the topical area of the selected clusters. This aids the user in subject focusing and topic/treatment discrimination.

6. It supports open-ended, exploratory browsing through following dynamically established linkages between records in the database, in order to retrieve materials related to those already found. These can be dynamically generated "hypersearches" that let users issue a Boolean query with a mouse click to find all items that share some field with a displayed record.

7. It uses the user's selection of relevant citations to refine the initial search statement and automatically construct new search statements for relevance feedback searching.

A primary goal of the Cheshire II system design was to provide an extensible system that can easily adapt to new types of data, and that could provide a flexible and programmable user interface to display that data. In order to achieve this goal, we have attempted to incorporate appropriate national and international standards into the system wherever possible.

As it turned out these goals were not as unique as they seemed during the initial design of the system, and apparently other had much the same notions of what sort of standards a new IR system should support. These other systems (such as the ZPrise system used as the control in the interactive track) share some of the characteristics of the Cheshire II system. There have been other online catalog systems that have provided ranked retrieval(Fox *et al.* 1993; Larson 1992; Robertson 1997; Porter 1988), and a number of systems provide Z39.50 access, and yet others provide SGML support. In the Cheshire system we have tried to keep pushing the edge of system development in including database techniques for index management, support for Z39.50 V.3 operations, and full SGML parsing.

1.1 The Cheshire II Search Engine

The Cheshire II search engine was designed to support a variety of search and browsing capabilities. We have included facilities for both probabilistic and Boolean searching in Cheshire II. This was driven by the realization that there are different types of search tasks that are best handled by different retrieval methods. Therefore, we provide support for such methods as authority-controlled name searching and other conventional online catalog search features, such as "exact title" and "exact subject" matching capability and the ability to store and retrieve both Boolean and probabilistic "result sets" and use them in subsequent queries.

The search engine also supports various methods for translating a searcher's query into the terms used in indexing the database. These methods include elimination of unused words using field-specific stopword lists, particular field-specific query-to-key conversion or "normalization" functions, standard stemming algorithms (Porter stemmer(Porter 1988)) and support for mapping database and query text words to single forms based on the WordNet dictionary and thesaurus using a adaption of the WordNet "Morphing" algorithm and exception dictionary..

However, the primary functionality that distinguishes the Cheshire II search engine is support for probabilistic searching on any indexed element of the database. This means that a natural language query can be used to retrieve the records that have of highest probability of being relevant given the user's query. In both cluster searching and direct probabilistic searching of the database, the Cheshire II search engine supports a very simple form of *relevance feedback*, where any items found in an initial search (Boolean or probabilistic) can be selected and used as queries in a relevance feedback search.

The system also supports the two-stage search method developed in the Cheshire prototype(Larson 1992). In the prototype probabilistic retrieval methods were used to match the searcher's query with a set of *classification clusters*, the searcher then selected the clusters that appeared relevant and they were combined with the initial query and used to re-rank the database, so that records were retrieved in decreasing order of probable relevance to the searcher's initial query statement combined with the broad classes selected in the first stage. This two-stage search method appeared to assist the searcher in subject focusing and topic/treatment discrimination(Larson 1991c). The cluster search method is still available in Cheshire II, but is now augmented by direct probabilistic searching of the database. This method was not used in the TREC-6 interactive track, although it is used with some success on other Cheshire databases.

1.1.1 Probabilistic Retrieval in Cheshire II

The probabilistic retrieval algorithm used in the Cheshire II search engine is based on the *logistic regression* algorithms developed by Berkeley researchers (Cooper *et al.* 1992; Cooper *et al.* 1994a; Cooper *et al.* 1994b). Formally, the probability of relevance given a particular query and a particular record in the database $P(R \mid Q, D)$ is calculated and the records are presented to the user ranked in order of decreasing values of that probability. In the Cheshire II system $P(R \mid Q, D)$ is calculated as the "log odds" of relevance $\log O(R \mid Q, D)$, where for any events A and B the odds $O(A \mid B)$ is a simple transformation of the probabilities $\frac{P(A|B)}{P(\overline{A}|B)}$. The Logistic Regression method provides estimates for a set of coefficients, c_i, associated with a set of S statistics, X_i, derived from the query and database, such that

$$\log O(R \mid Q, D) \approx c_0 \sum_{i=1}^{S} c_i X_i \tag{1}$$

where c_0 is the intercept term of the regression.

For the set of M *terms* (i.e., words, stems or phrases) that occur in both a particular query and a given document. The equation used in estimating the probability of relevance for the Cheshire II search engine is essentially the same as that used in (Cooper *et al.* 1994b) where the coefficients were estimated using relevance judgements from the TIPSTER test collection:

$X_1 = \frac{1}{M} \sum_{j=1}^{M} logQAF_{t_j}$. This is the log of the absolute frequency of occurrence for term t_j in the query averaged over the M terms in common between the query and the document. The coefficient c_1 used in the current version of the Cheshire II system is 1.269.

$X_2 = \sqrt{QL}$. This is square root of the query length (i.e., the number of terms in the query disregarding stopwords). The c_2 coefficient used is -0.310.

$X_3 = \frac{1}{M} \sum_{j=1}^{M} logDAF_{t_j}$. This is is the log of the absolute frequency of occurrence for term t_j in the document averaged over the M common terms. The c_3 coefficient used is 0.679.

$X_4 = \sqrt{DL}$. This is square root of the document length. In Cheshire II the raw size of the document in bytes is used for the document length. The c_4 coefficient used is -0.0674.

$X_5 = \frac{1}{M} \sum_{j=1}^{M} logIDF_{t_j}$. This is is the log of the *inverse document frequency*(IDF) for term t_j in the document averaged over the M common terms. IDF is calculated as the total number of documents in the database, divided by the number of documents that contain term t_j The c_5 coefficient used is 0.223.

$X_6 = logM$. This is the log of the number of common terms. The c_6 coefficient used in Cheshire II is 2.01.

These coefficients and elements of the ranking algorithm have proven to be quite robust and useful across a broad range of document types.

The Cheshire II search engine calculates all matching functions at the point of retrieval, rather than pre-computing portions of the functions. Only the fundamental statistics (such as raw term frequency) are stored in the database, making it easy to apply a different algorithm to the same database without re-indexing, and simplifying incremental updates and additions to the database.

Probabilistic searching, as noted above, requires only a natural language statement of the searcher's topic, and thus no formal query language or Boolean logic is needed for such searches. However, the Cheshire II search engine also supports complete Boolean operations on indexed elements in the database, and supports searches that combine probabilistic and Boolean elements. At present, combined probabilistic and Boolean search results are evaluated using the assumption that the Boolean retrieved set has an estimated $P(R \mid Q_{bool}, D) = 1.0$ for each document in the set, and 0 for the rest of the collection. The final estimate for the probability of relevance used for ranking the results of a search combining Boolean and probabilistic strategies is simply:

$$P(R \mid Q, D) = P(R \mid Q_{bool}, D) P(R \mid Q_{prob}, D) \tag{2}$$

where $P(R \mid Q_{prob}, D)$ is the probability estimate from the probabilistic portion of the search, and $P(R \mid Q_{bool}, D)$ the estimate from the Boolean. This has the effect of restricting the results to those items that match the Boolean portion, with ordering based on the probabilistic portion.

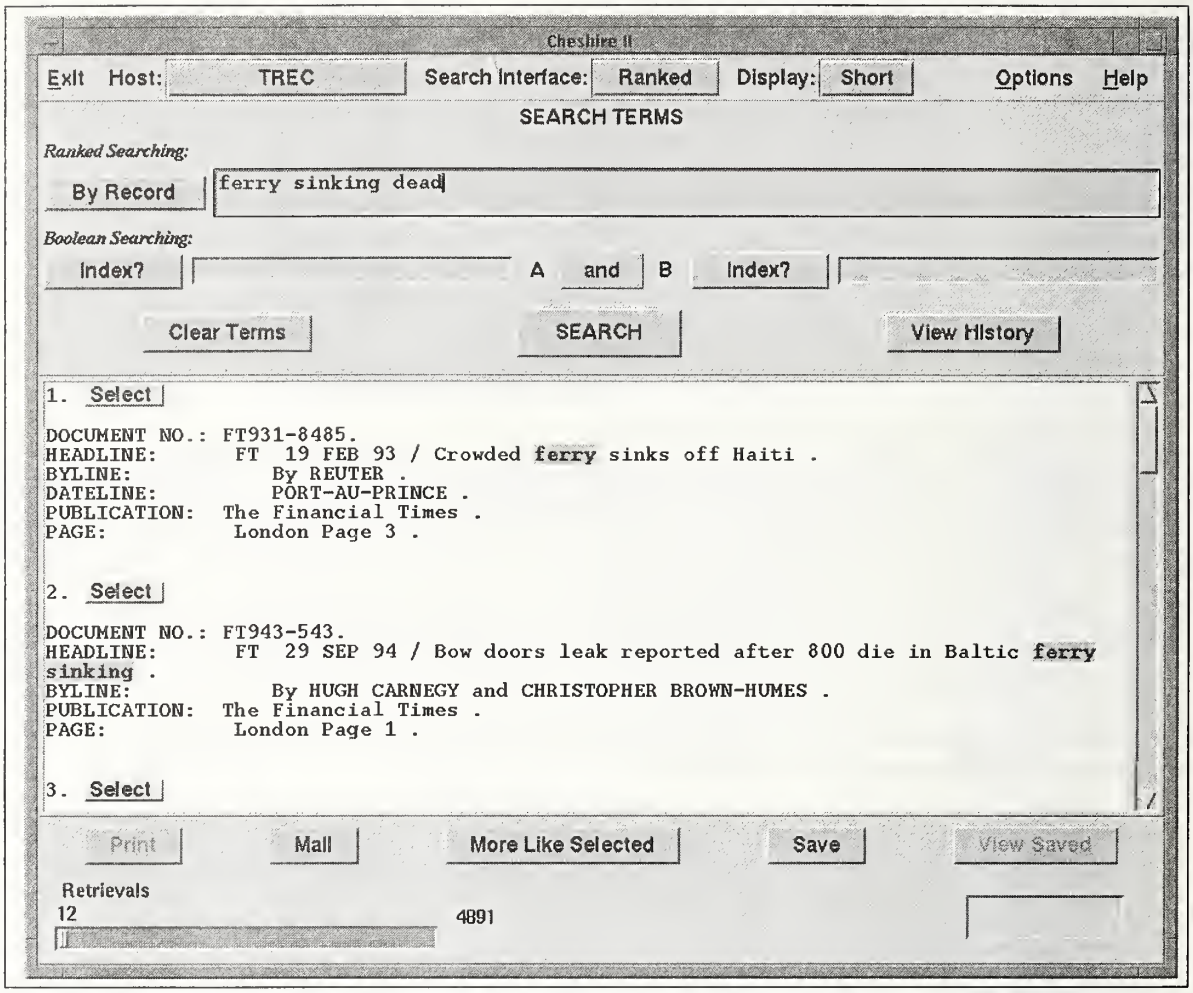


Figure 1: Cheshire II Client: Short Display Format

1.1.2 Relevance Feedback

In the current implementation of the Cheshire II system, relevance feedback is implemented quite simply, as probabilistic retrieval based on extraction of content-bearing elements (such as titles, subject headings, etc.) from any items that have already been seen and selected by a user. Thus, any citation or document seen by the user can become the basis for a *nearest neighbor* search, where it is used as a query to find those records in the database most similar in content to the one specified. Similarly, multiple records may be selected and submitted for feedback searching. In this case the contents of all those records are merged into a single query and submitted for searching. In the current implementation, generating a feedback search is accomplished by parsing the selected record(s) and extracting the record elements specified for the index used for topical searching (as specified in the database configuration file). Each of these record elements is combined to form a single query, which is then submitted to the same probabilistic retrieval process described above. At the present time we do not use any methods for eliminating poor search terms from the selected records, nor special enhancements for terms common between multiple selected records(Salton & Buckley 1990), but we plan to experiment further with various enhancements to our relevance feedback method.

1.2 The Cheshire II Client Interface

The design of the Cheshire II client interface (show with the TREC FT database in Figures 1 and 2, has been driven by a number of goals:

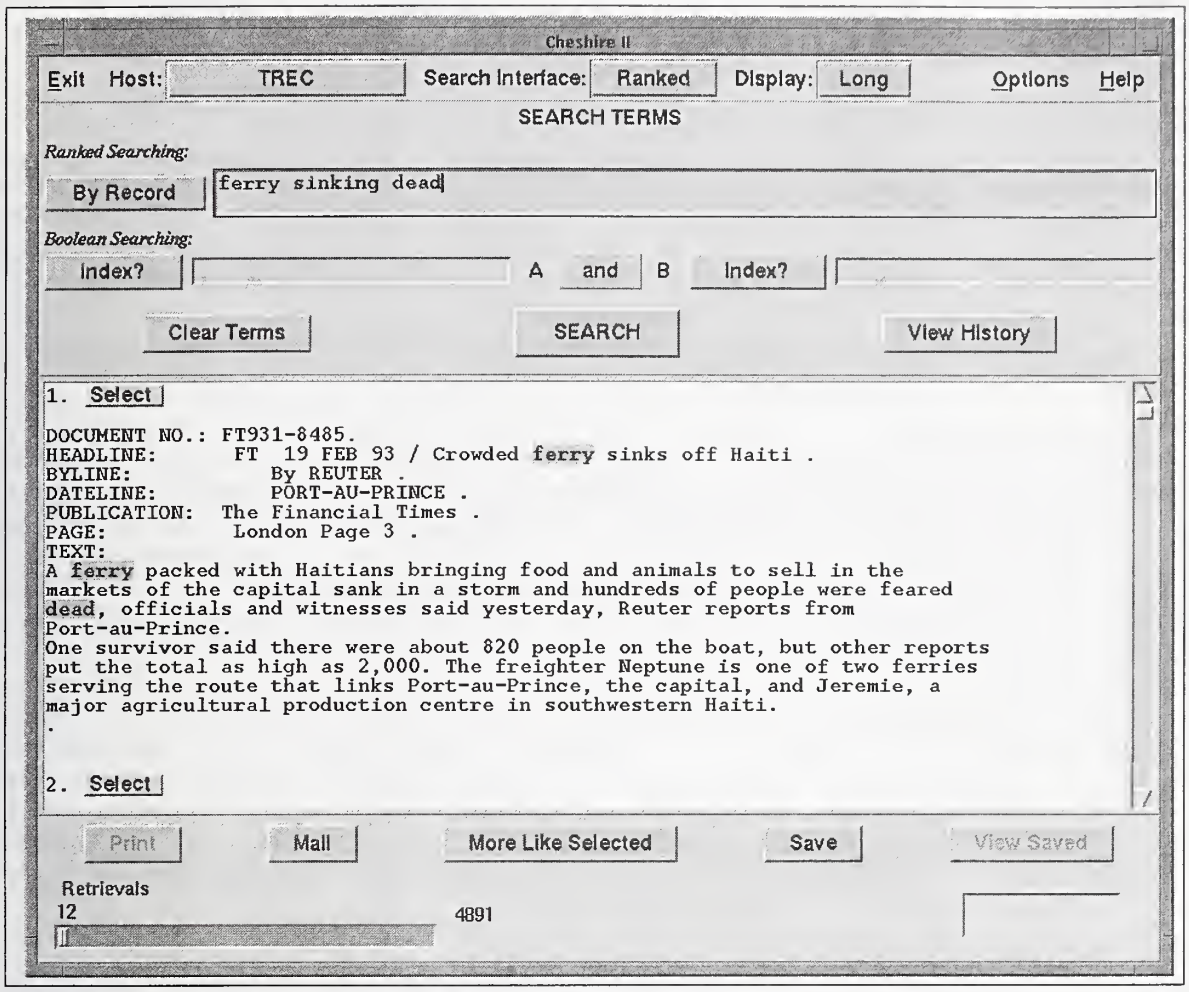


Figure 2: Cheshire II Client: Long Display Format

1. to support a consistent interface to a wide variety of Z39.50 servers, and to dynamically adapt to the particular server.

2. to reduce the cognitive load on the users wishing to interact with multiple distributed information retrieval systems by providing a single interface for them all.

3. to minimize use of additional windows during users' interactions with the client in order to allow them to concentrate on formulating queries and evaluating the results, and not expend additional mental effort and time switching their focus of attention from the search interface to display clients;

4. to provide functions not immediately related to searching, such as print and e-mail facilities, to facilitate users' ability to 'take the results home'; and

5. to design a help system within the interface that would assist users not only in the mechanics of operating the Cheshire II client, but also in the more general tasks of selecting appropriate resources for searching, formulating appropriate queries, and employing various search tactics.

Additional functionality beyond searching and browsing has been relatively easy to implement. Functions for printing, e-mailing and saving records are all available when records are displayed, and the user has the option of acting on either the entirety of the current record display or a subset thereof by selecting individual records using the "select" buttons on each record (visible in Fig. 1 next to the record numbers).

The Cheshire II client interface has been primarily implemented using the interpreted Tcl/Tk language(Ousterhout 1994), with a variety of lower-level functions, including the majority of the Z39.50 client

interactions, written in the C programming language. This combination has proven quite successful in both providing the ability to rapidly prototype and modify the graphic user interface to accommodate new features, and to maintain a relatively high level of performance for the Z39.50 client-server interactions.

In addition to the Cheshire II client interface, complete access to the Cheshire II server is available through other Z39.50 clients. The Cheshire II server also provides support for the HTTP protocol via an HTTP-to-Z39.50 gateway, giving access to popular WWW clients like Netscape and Internet Explorer. This interface (using HTML forms for data entry elements) provides remote network users many of the same search features as the full client described above, with some loss of integration and ease of interactivity. Because HTTP is a stateless protocol, with each query/response pair considered a complete transaction, the ability to do relevance feedback is very limited in the current WWW implementations.

2 TREC Interactive Track

Because this is the first time the Cheshire II system has been used in the TREC tasks, we made a number of changes and additions to the system, primarily to more efficiently support the characteristics of the Financial Times(FT) database. Athough the 600Mb of FT data was actually smaller than some of our existing databases, such as the NSF/NASA/ARPA Digital Library database with over 200,000 pages (approx 1Gb) of OCRed full-text, it was much "cleaner" and more consistent in spelling and language usage, leading to larger postings lists than we had previously encountered. The changes included modifications to the structure of the Cheshire indexes and support for more modern database technology in our B-trees and in index loading. However, the fundamental ranking and retrieval mechanisms described above were not modified for TREC.

Additional changes were made in the user interface, where display formats for the FT records were designed (as shown in Fig. 1 and Fig. 2), and a routine was added to highlight query terms in the text of the document to aid searchers in scanning for relevant passages. Note that the highlighting feature doesn't necessarily catch all of the terms that contributed to the selection of the document, because only the original query terms, and not stemmed terms, are used in the highlighting. (This may be seen in Fig. 2, where the title term "sinks" is not highlighted because it is not an exact match of the query term "sinking").

2.1 User Characteristics

The administration of the interactive track followed the track guidelines with a single group of 4 participants. While none of the participants had used either the experimental (Cheshire II) or control (XPRISE) systems in searching tasks, they had all seen demonstrations of the experimental system. All of the participants held college degrees (three were PhD candidates). Two of the participants (P1 and P3) had considerable experience in online searching on other systems, the other two had very limited experience with online systems.

2.2 Search Results

Table 1 shows the differences between the Cheshire II system and the global average for the control system at all participating sites. The aspectual recall results for the Cheshire II system were the highest of all the participating sites (see the Interactive Track overview), while still maintaining a fairly high level of aspectual precision.

Table 2 shows the frequency and percentages of relevant documents (i.e. documents containing one or more relevant aspects) found by the different participants. As the column totals suggest, the experienced searchers (P1 and P3) were much more likely to find more of the relevant documents (61.37%). This appeared to be mostly due to more persistence in searching, and trying different search strategies, compared to the inexperienced searcher. As Tables 3 and 4 show, higher numbers of relevant documents were found by the experienced searchers in both the experimental system and the control system. There were, of course, substantial differences between individual searchers in performance on each search. For example, searcher P2 outperformed the experienced searcher P3 using the Cheshire system, and outperformed the experienced searcher P1 using the ZPRISE system, even though the combined score was smaller than either of the combined scores for the experienced searchers.

Table 5 shows the aspectual precision and recall as calculated by NIST for each of the searchers and queries. One particularly interesting observation was that in the three searches (303i, 339i, 347i) where the inexperienced users (P2 and P4) used Cheshire II and the experienced users used ZPRISE, in most cases

Measure	Mean	Std Dev
Aspectual Recall	0.079	0.102
Aspectual Precision	-0.012	0.155
Time	94.750	121.589

Table 1: Berkeley Cheshire II Difference from Global Control

the aspectual recall for inexperienced users matched or even exceeded the performance of the experienced searchers. This was a very gratifying result since one of the initial design goals for the cheshire system was to improve retrieval performance for inexperienced users.

3 Conclusions

Naturally, it it impossible to draw any general conclusions from our small sample size. However, the overall performance of the Cheshire II system seemed fairly good, although it was not dramatically better than the control system in most cases. The results, as has often been noted in previous TREC interactive evaluations, tend to be highly influenced by individual behavior and search techniques (this is apparent in the differences between the experienced searchers on the same questions and in the same systems). While we had hoped to be able to see Cheshire might be more effective for the inexperienced searchers, the results are ambiguous.

Of particular interest for further examination is why some searches seemed to be done more effectively on the control system, and some on the Cheshire system. The likely answer is that the differences in ranking mechanisms provide advantages in certain situations. We plan to explore these situations with an eye towards improving the Cheshire II ranking algorithms.

4 Acknowledgements

The development of the Cheshire II system was sponsored by a College Library Technology and Cooperation Grants Program, HEA-IIA, Research and Demonstration Grant (#R197D30040) from the U.S. Department of Education. Ongoing work on the Cheshire II project and system is supported as part of Berkeley's NSF/NASA/ARPA Digital Library Initiative Grant #IRI-9411334.

References

Cooper, W. S., Gey, F. C., & Dabney, D. P. (1992). Probabilistic Retrieval Based on Staged Logistic Regression. In: *SIGIR '92 (Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, June 21-24, 1992)* (pp. 198-210). New York: ACM.

Cooper, W. S., Gey, F. C. & Chen, A. (1994a). Full Text Retrieval based on a Probabilistic Equation with Coefficients fitted by Logistic Regression. In: D. K. Harman (Ed.) *Second Text Retrieval Conference (TREC-2), Gaithersburg, MD, USA, 31 Aug.-2 Sept. 1993*, NIST-SP 500-215, (pp. 57-66). Washington : NIST.

Cooper, W. S., Chen, A. & Gey, F. C. (1994b). Experiments in the Probabilistic Retrieval of Full Text Documents In: *Text Retrieval Conference (TREC-3) Draft Conference Papers,* Gaithersburg, MD : National Institute of Standards and Technology.

Fox, E. A., France, R. K., Sahle, E., Daoud, A. & Cline, B. E. (1993). Development of a Modern OPAC: From REVTOLC to MARIAN. IN: *SIGIR '93 (Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, June 27-July 1, 1993)* (pp. 248-259). New York: ACM.

Larson, R. R. (1991c). Classification Clustering, Probabilistic Information Retrieval, and the Online Catalog. *Library Quarterly*, 61, 133-173.

Query ID	*User Number*				
Frequency *Percentage* *Row Pct.* *Col. Pct.*					
	P1	P2	P3	P4	Total
303i	3	2	5	5	15
	2.07	1.38	3.45	3.45	10.34
	20.00	13.33	33.33	33.33	
	6.82	6.67	11.11	19.23	
307i	17	9	9	6	41
	11.72	6.21	6.21	4.14	28.28
	41.46	21.95	21.95	14.63	
	38.64	30.00	20.00	23.08	
322i	8	3	5	1	17
	5.52	2.07	3.45	0.69	11.72
	47.06	17.65	29.41	5.88	
	18.18	10.00	11.11	3.85	
326i	8	3	6	4	21
	5.52	2.07	4.14	2.76	14.48
	38.10	14.29	28.57	19.05	
	18.18	10.00	13.33	15.38	
339i	4	4	7	4	19
	2.76	2.76	4.83	2.76	13.10
	21.05	21.05	36.84	21.05	
	9.09	13.33	15.56	15.38	
347i	4	9	13	6	32
	2.76	6.21	8.97	4.14	22.07
	12.50	28.13	40.63	18.75	
	9.09	30.00	28.89	23.08	
Total	44	30	45	26	145
	30.34	20.69	31.03	17.93	100.00

Table 2: Relevant Documents by User and Query

Query ID	*User Number*				
Frequency *Percentage* *Row Pct.* *Col. Pct.*					
	P1	P2	P3	P4	Total
303i	0	2	0	4	6
	0.00	3.13	0.00	6.25	9.38
	0.00	33.33	0.00	66.67	
	0.00	13.33	0.00	30.77	
307i	13	0	6	0	19
	20.31	0.00	9.38	0.00	29.69
	68.42	0.00	31.58	0.00	
	56.52	0.00	46.15	0.00	
322i	3	0	1	0	4
	4.69	0.00	1.56	0.00	6.25
	75.00	0.00	25.00	0.00	
	13.04	0.00	7.69	0.00	
326i	7	0	6	0	13
	10.94	0.00	9.38	0.00	20.31
	53.85	0.00	46.15	0.00	
	30.43	0.00	46.15	0.00	
339i	0	4	0	4	8
	0.00	6.25	0.00	6.25	12.50
	0.00	50.00	0.00	50.00	
	0.00	26.67	0.00	30.77	
347i	0	9	0	5	14
	0.00	14.06	0.00	7.81	21.88
	0.00	64.29	0.00	35.71	
	0.00	60.00	0.00	38.46	
Total	23	15	13	13	64
	35.94	23.44	20.31	20.31	100.00

Table 3: Relevant Documents by User and Query: Cheshire Only

Query ID	*User Number*				
Frequency *Percentage* *Row Pct.* *Col. Pct.*					
	P1	P2	P3	P4	Total
303i	2	0	3	0	5
	4.00	0.00	6.00	0.00	10.00
	40.00	0.00	60.00	0.00	
	20.00	0.00	16.67	0.00	
307i	0	8	0	6	14
	0.00	16.00	0.00	12.00	28.00
	0.00	57.14	0.00	42.86	
	0.00	66.67	0.00	60.00	
322i	0	2	0	1	3
	0.00	4.00	0.00	2.00	6.00
	0.00	66.67	0.00	33.33	
	0.00	16.67	0.00	10.00	
326i	0	2	0	3	5
	0.00	4.00	0.00	6.00	10.00
	0.00	40.00	0.00	60.00	
	0.00	16.67	0.00	30.00	
339i	4	0	4	0	8
	8.00	0.00	8.00	0.00	16.00
	50.00	0.00	50.00	0.00	
	40.00	0.00	22.22	0.00	
347i	4	0	11	0	15
	8.00	0.00	22.00	0.00	30.00
	26.67	0.00	73.33	0.00	
	40.00	0.00	61.11	0.00	
Total	10	12	18	10	50
	20.00	24.00	36.00	20.00	100.00

Table 4: Relevant Documents by User and Query – ZPRISE Only

Searcher	Topic	System	num. docs	Prec	Rec.
P1	307i	CHESHIRE	17	0.765	0.565
P1	322i	CHESHIRE	8	0.375	0.222
P1	326i	CHESHIRE	8	0.875	0.667
P1	303i	ZPRISE	3	0.667	1.000
P1	339i	ZPRISE	4	1.000	0.800
P1	347i	ZPRISE	4	1.000	0.154
P2	303i	CHESHIRE	2	1.000	1.000
P2	339i	CHESHIRE	4	1.000	0.900
P2	347i	CHESHIRE	9	1.000	0.308
P2	307i	ZPRISE	9	0.889	0.348
P2	322i	ZPRISE	3	0.667	0.222
P2	326i	ZPRISE	3	0.667	0.333
P3	307i	CHESHIRE	9	0.667	0.261
P3	322i	CHESHIRE	5	0.200	0.111
P3	326i	CHESHIRE	6	1.000	0.667
P3	303i	ZPRISE	5	0.600	1.000
P3	339i	ZPRISE	7	0.571	0.900
P3	347i	ZPRISE	13	0.846	0.462
P4	303i	CHESHIRE	5	0.800	1.000
P4	339i	CHESHIRE	4	1.000	0.900
P4	347i	CHESHIRE	6	0.833	0.269
P4	307i	ZPRISE	6	1.000	0.261
P4	322i	ZPRISE	1	1.000	0.111
P4	326i	ZPRISE	4	0.750	0.333

Table 5: Aspectual Precision and Recall by Searcher, Query, and System

Larson, R. R. (1992). Evaluation of Advanced Retrieval Techniques in an Experimental Online Catalog. *Journal of the American Society for Information Science*, 43, 34-53.

Larson, R. R., McDonough, J., O'Leary, P., Kuntz, L. & Moon, R. (1996). Cheshire II: Designing a Next-Generation Online Catalog. *Journal of the American Society for Information Science*, 47, 555-567.

Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit* Reading, Mass. : Addison-Wesley.

Porter, M. & Galpin, V. (1988). Relevance feedback in a public access catalogue for a research library: Muscat at the Scott Polar Research Institute. *Program*, 22, 1-20.

Robertson, S.E. (1997). Overview of the Okapi projects. *Journal of Documentation*, 53(1), 3-7.

Salton, G. & Buckley, C. (1990). Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41, 288-297.

Turtle, H. R.& Croft, W. B. (1990). Inference Networks For Document Retrieval. In: J. Vidick (Ed.)*Proceedings of the 13th International Conference on Research and Development in Information Retrieval. (Proceedings of the 13th International Conference on Research and Development in Information Retrieval, Brussels, Belgium, 5-7 Sept. 1990)* (pp. 1-24). New York: ACM.

Fusion Via Linear Combination for the Routing Problem

Christopher C. Vogt, Garrison W. Cottrell
University of California, San Diego
Computer Science and Engineering 0114
La Jolla, CA 92093
{vogt,gary}@cs.ucsd.edu

February 6, 1998

Abstract

A linear combination of scores from two different IR systems is used for the routing task, with one combination model being trained for each query. Despite a poor selection of component systems, the combination model performs on par with the better of the two systems, learning to ignore the worse system.

1 INTRODUCTION

Our work this year followed up on our TREC5 fusion approach – a linear combination of relevance scores (a.k.a. RSVs) from different IR systems. Last year's *adhoc* entry successfully improved performance over all three component systems. However, our *routing* entry did not show an improvement, but rather a degradation in performance when compared to the best individual system. We attributed these disappointing results to three possible factors: overfitting, a weak combination model, or the fact that we used a single set of model parameters for all routing queries rather than training a separate model to each query. This year's entry addressed the last factor by using the same linear model and similar training method, but customizing an individual model for each query.

Our participation was in the Category A, routing task.

2 METHOD

A linear combination model is used to compute the weighted sum of scores from two IR systems. Since we are in the routing context, we can effectively

ignore the query as an input, thus the score R for a particular document d on a particular query q is computed as:

$$R_q(w, d) = R_{q,1}(d) + wR_{q,2}(d) \quad (1)$$

A single weight w is used instead of two, because all that matters is the *ranking* of documents, and thus only the ratio of weights. w is scanned from 20 to $\frac{1}{20}$ in multiplicative increments of 0.95, with scores from each system pre-normalized by dividing by the respective averages. This normalization allows the above technique of scanning the weight to effectively cover all possible different combinations even though it only covers a small interval of possible weights. Negative weights were not examined. The w which maximizes average precision on the training set is selected for each query. Our entry into last year's TREC optimized a different criterion, J, which measures how close the combined system's ranking is to the user's, and is highly correlated with average precision. Since we have only one parameter in our model this year, it was computationally feasible to optimize average precision directly.

The two "systems" used were both variants based on Verity Inc.'s routing submissions (due to the first author's affiliation with Verity over the summer). These systems make use of Verity's "Query By Example" (QBE) functionality, which generates a query in Verity's rich VQL query language based on positive and negative document examples. The first system used was a version trained primarily on documents from the same source as the test set (FBIS) which varied how many of the top-ranked documents and terms from these documents were used to construct the query, choosing different numbers of examples and terms for each query. The second system used a constant number of terms (15) and used all possible positive training examples (regardless of which collection they came from) and no negative examples, regardless of the query. We will refer to first system as the source-specific system because of its emphasis on FBIS documents, and the second as the fixed system, because it did not optimize the parameters (number of documents and terms) of the QBE query generator.

3 RESULTS and DISCUSSION

Figure 1 shows the precision/recall graph for each individual system and the combined system. Verity's run is included for comparison because it also used a fusion approach. Verity's approach was to choose whichever of two different systems performed better on the training set on a per query basis. One of the two systems Verity used was the same as our source-specific system. Also shown are results for the best possible fusion using the linear model, found by optimizing the weight using the *test* data. This indicates an upper bound on achievable performance.

The interesting points to note about the graph are:

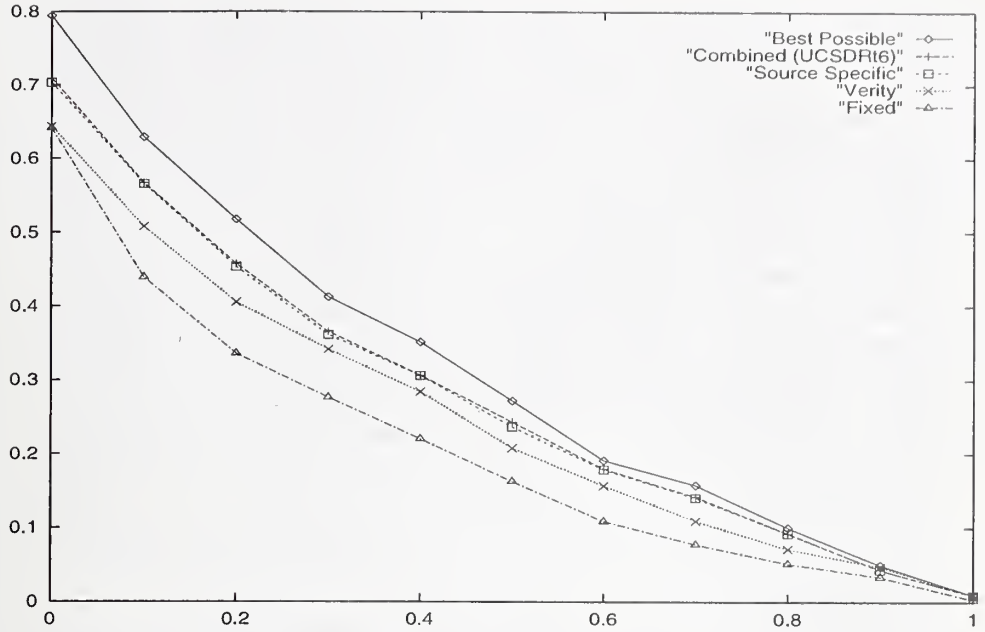


Figure 1: Precision/Recall Graph of Test Set Performance

1. The combination system has performance virtually identical to the better (source-specific) of the two systems.

2. The combination system does not achieve the best possible linear performance (given knowledge of the test set).

3. Verity's fusion did not fare as well as ours, underperforming the better of its two component systems (the source-specific system).

The first point is disappointing, since previous work ([Belkin et al., 1995], [Lee, 1997], [Bartell et al., 1994], etc.) has shown that the combination should *outperform* the best individual system. Closer examination of the model reveals why. Examination of the weight used in the combination for each query shows that in 29 of the 47 queries (62%), the fixed system received *zero* weight. Furthermore, in 76% of the queries, the fixed system's contribution was less than 10%, and for all but one of the queries, the source-specific system was weighted more heavily. Considering this, the similar performance of the source-specific system and the combined system is not surprising, but raises the question of why one system is always weighted more heavily than the other.

Work which we have done concurrently ([Vogt, 1997]) may shed some light on this – it appears that the problem is which systems we chose to combine. Our work indicates that the best time to linearly combine systems is when they

a) both have performance of similar magnitudes and b) rank relevant documents differently. For our training data, nearly 80% of the queries exhibited a difference in magnitude of performance (average precision) of 0.1 or more, with the average being 0.2. Of those 10 queries in which both systems did exhibit similar performance, only 2 ranked relevant documents differently[1]. Thus, it appears we were attempting to combine systems which had little potential for improvement.

The best possible combination line shown in Figure 1 shows that a linear model could indeed significantly outperform the source-specific system, with an average precision of 0.32 versus 0.28. However, it's doubtful that this combination is achievable using the available training data. In fact, on the training set, the source-specific system had average precision of 0.40 and the mixture weighed in at 0.41, a very small difference.

However, we note that our model and training technique apparently were able to generate a combination which was at least as good as the better system. In fact, on all but one query, the combined system's performance on the test set was identical to the source-specific system's. Thus, our training technique was able to recognize that the fixed system generally could not contribute much and therefore to ignore it.

The precision/recall graph shows that our technique is better than the system selection approach used by Verity, which achieved performance somewhere between its two component systems (Verity's second system is not shown, but performs slightly better than the fixed system). However, as Verity points out in its TREC report, their fusion approach was one which has not generally done well in the past (system selection), so perhaps this comparison is not very informative.

4 CONCLUSIONS and FUTURE WORK

Our results show that, unlike last year's entry, training one model per query results in a system at least as good as the best expert. However, no major improvement over the best expert was obtained. Again, we believe this was due to combining a good expert with a poor one, and since this technique has generally proven effective for other IR researchers, we maintain interest in pursuing this approach.

The linear combination model is theoretically capable of performing much better than our entry did (about 14% at low recall levels). This may be due to insufficient training data, outliers, or an inadequate training method. For example, we optimized performance on the training set, rather than using a hold–out set to stop training, a technique which should give us better generalization. This approach, and training on only the top-ranked documents to

[1] As measured by GPA_r, the Guttman's Point Alienation calculated using only relevant documents – see [Vogt, 1997]

avoid outliers, are two techniques we are currently investigating. Several other issues, such as the use of negative weights and score normalization are also part of ongoing research on the linear model.

As noted above, the inability to improve on the better system may be due to the particular systems we chose to combine. Our ongoing work is investigating this by looking at combinations of a broad spectrum of different IR systems (the actual entries from past TRECs), thus allowing more serendipitous combinations to be found. In addition to the linear model, we are investigating neural network models which are capable of implementing a broader range of combination functions and taking query and document representations into account.

Perhaps the most interesting conclusion from this work is that our training technique this year, which included per-query training, was robust to a poor selection of component systems. Given a pair of experts which were unlikely to be combinable, the training process was able to identify the "bad" system and ignore it.

5 Acknowledgments

The authors would like to thank Dominic Lobbia for all the hard work he did in the initial stages of our work for TREC.
This research was supported by NSF grant IRI 92-21276.

References

[Bartell et al., 1994] Bartell, B. T., Cottrell, G. W., and Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. In Croft, W. B. and van Rijsbergen, C., editors, *SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin. Springer-Verlag.

[Belkin et al., 1995] Belkin, N., Kantor, P., Fox, E., and Shaw, J. (1995). Combining evidence of multiple query representations for information retrieval. *Information Processing and Management*, 31(3):431–448.

[Belkin et al., 1997] Belkin, N. J., Narasimhalu, A. D., and Willett, P., editors (1997). *SIGIR 97: Proceedings of the Twentieth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia. ACM Press.

[Lee, 1997] Lee, J. H. (1997). Analyses of multiple evidence combination. In [Belkin et al., 1997], pages 267–276.

[Vogt, 1997] Vogt, C. C. (1997). When does it make sense to linearly combine relevance scores? In [Belkin et al., 1997]. Poster Session. UCSD CSE Tech Report CS97-556.

Short Queries, Natural Language and Spoken Document Retrieval: Experiments at Glasgow University

Fabio Crestani*, Mark Sanderson†, Marcos Theophylactou, Mounia Lalmas
Department of Computing Science
University of Glasgow
Glasgow G12 8QQ, Scotland

Abstract

This paper contains a description of the methodology and results of the three TREC submissions made by the Glasgow IR group (glair). In addition to submitting to the ad hoc task, submissions were also made to NLP track and to the SDR speech 'pre-track'. Results from our submissions reveal that some of our approaches have performed poorly (i.e. ad hoc and NLP track), but we have also had success particularly in the speech track through use of transcript merging. We also highlight and discuss a seemingly unusual result where retrieval based on the very short versions of the TREC ad hoc queries produced better retrieval effectiveness than retrieval based on more 'normal' length queries.

1 Introduction

This paper contains a description of the methodology and results of the ad hoc, NLP, and SDR submissions made by the Glasgow IR group (**glair**) to this year's TREC. The only common factor between the submissions is their

*Supported by a "Marie Curie" Research Fellowship from the European Union.
†Supported by VIPIR project of the University of Glasgow

use of a Glasgow built retrieval system, SIRE and this is introduced first in the paper. As the submissions are quite independent of each other, the rest of the paper is structured as an amalgam of three sub papers each with their own introduction, methodology, results and conclusions. The order of these sub papers is first, the ad hoc submission, second the NLP track, and finally the SDR track submission.

2 The SIRE Information Retrieval system

The system used in in the context of the work reported in this paper is a retrieval toolkit called *SIRE* (System for Information Retrieval Experimentation) developed "in-house" at Glasgow University by Mark Sanderson. SIRE is a collection of small independent modules, each conducting one part of the indexing, retrieval and evaluation tasks required for classic retrieval experimentation. The modules are linked in a pipeline architecture communicating through a common token based language. SIRE was initially used in research examining the relationship between word sense ambiguity, disambiguation, and retrieval effectiveness [8]. It proved to be a flexible tool as it not only provided retrieval functionality but a number of its core modules were used to build a word sense disambiguator as well. It was also used in the experiments for the Glasgow IR group submissions to TREC-4 and TREC-5 and is currently being used in a number of research efforts within the group.

SIRE is implemented on the UNIX operating system which, with its scripting and pre-emptive multi-tasking is eminently suitable for handling the modular nature of SIRE.

SIRE was chosen as the IR platform for the experiments reported in this paper because it implemented a probabilistic IR model we are very familiar with, based on the "TF-IDF" weighting schema [12]. Moreover, it was relatively easy to modify the code to take into account the characteristics of the new data.

A detailed description of the functionalities of SIRE is outside the scope of this paper. The system is currently public available for research purposes. The interested reader should contact Mark Sanderson for a copy of a short unpublished paper describing the system [7] and for the location of SIRE's binary files. The system has been successfully used by many students of the Advanced Information Systems M.Sc. of Glasgow University for their practical work.

3 Main ad hoc task: short queries and semi-automatic query expansion

In the ad hoc task of TREC the Glasgow IR group submitted three runs: **glair61**, **glair62**, and **glair64**. The main aim of this work was to investigate a means of improving retrievals for the very short queries of TREC-6. Because of their length, it was assumed that their use would result in poor retrieval and it would be necessary to expand them in some manner. The first two submissions (**glair61** & **glair62**) were aimed at testing such an expansion technique based on the manual identification of the senses of query words and the subsequent automatic expansion of those senses.

This work was somewhat overshadowed by the effectiveness results returned from the **glair64** submission - retrieval based on normal length queries (i.e. TREC query description fields) - which proved to be worse than the **glair61** results - retrieval based on the very short queries (i.e. their title fields). In other words, the very short queries were better than the normal length queries.

The rest of this section will first, describe the implementation, and results of the semi-automatic query expansion experiments and second, explore possible reasons for the drop in retrieval effectiveness found to occur when using the longer, and presumably more detailed, versions of the TREC queries.

3.1 Semi-automatic query expansion

A new feature of TREC this year was the introduction of the very short query task: ad hoc retrieval based on the title section of TREC queries. These queries were intended to mimic the type of query normally submitted to interactive IR systems by untrained, casual users. Their generation was governed by a set of guidelines[9], an extract of which is shown below.

> ... we would like you to make an effort in ensuring that the length of the titles is kept as short as possible. Please try to keep the length of the title to between 1 and 3 non-stop words. Only in exceptional circumstances would they be any longer, for example, if the title were some well known phrase or a long proper name. Do not worry if the title is not an accurate expression of the information need, this is a common feature of very short

queries: there is only so much that can be expressed in such a small number of words.

The very short queries generated from these guidelines were on average 2.5 non-stop words in length, as opposed to the normal length queries (based on the description field) which were 8.5 non-stop words in length. Figure 1 shows a couple of these queries (numbers 310 & 349) to illustrate these two query types.

```
<title> Radio Waves and Brain Cancer

<desc> Description:
Evidence that radio waves from radio towers or car phones
affect brain cancer occurrence.


<title> Metabolism

<desc> Description:
Document will discuss the chemical reactions necessary to
keep living cells healthy and/or producing energy.
```

Figure 1: Queries 310 & 349

It would probably be fair to say that there was an assumption among many involved in the decision to include these queries in TREC-6 that the effectiveness of any IR system retrieving from them would be poor when compared to retrievals using the more normal TREC queries based on the description field. With this preconception in mind, it was decided (by one of the authors) to explore the possibility of incorporating some type of query expansion into the very short queries. The one chosen was a semi-automatic form that required the manual identification of the sense of each query word followed by the automatic expansion of the identified senses with synonyms taken from a thesaurus. Similar ideas of mixing manual tagging with thesaurus based expansion have been reported by [13]. One of the conclusions drawn from this research was that expansion of shorter queries was more likely to improve retrieval effectiveness than expansion of longer queries. It was hoped that this situation would be encountered in the experiments on the very short queries of TREC. However, another conclusion of [13] was that use of automatic expansion methods could make queries decidedly worse. It was hoped that

trying different forms of expansion in our experiments could counter these potential problems.

3.1.1 Implementation of experiments

There were three main components to this experiment: the document collection used, the retrieval system employed; and the thesaurus chosen to provide the sense definitions and synonyms. The collection was the 'A' collection as defined in the TREC-6 guidelines. The retrieval system employed was SIRE using standard IR features such as stop word removal, stemming and a $tf \times idf$ weighting scheme. The thesaurus used was WordNet [5], chosen because of it's coverage, ease of use and availability.

The first part of the expansion process involved the manual identification of query word senses. This was undertaken by one of the authors who looked up each query word in WordNet and assigned the sense closest to that word (this also involved the identification of the grammatical form that each word was used in). As WordNet stores phrases as well as words (e.g. 'land mine'), any possible query phrases were looked up before individual words were. Expansion of the word senses was simply a process of adding to the query exact synonyms of the senses. WordNet is quite sparing in its provision of synonyms, consequently queries were only expanded by a few words.

In choosing the precise form of expansion strategy employed for the TREC submission, experiments were run using the titles of the previous year's TREC queries (i.e. 251–300) on the 'B' collection of TREC-5. Results from these queries were disappointing: every expansion strategy tried was found to result in queries that produced lower retrieval effectiveness than that resulting from the unexpanded queries. Consequently, the 'least worst' strategy was chosen for submission in a vain hope that it would prove to be effective on the TREC-6 queries. The strategy consisted of expanding only the nouns of query words and leaving phrases unexpanded. In the experiments on queries 251–300, this strategy was found to improve 8 queries, leave 14 unchanged, and degrade 23 (the remaining 5 queries have no relevant documents). Unfortunately, this drop in effectiveness was repeated in the results returned from this year's TREC submission. The retrieval effectiveness of the queries after being expanded (**glair62**) was worse than the effectiveness of the unexpanded queries (**glair61**): with the expansion improving 3 queries, leaving 23 the same, and degrading 24.

As a footnote to this experiment, after submitting to TREC, some further

expansion strategies were attempted on the 251–300 queries and a strategy was found that improved upon previous strategies, though still caused a drop in effectiveness, albeit a small one. The strategy was motivated from work reported by [8] which showed how the frequency of occurrence of the senses of words was skewed so that the most common sense of a word typically accounted for the majority of occurrences of that word. With this information in mind, it was surmised that query words used in their commonest sense did not need expansion as their sense would be so prevalent in the collection, expansion terms would more likely introduce error than help retrieve documents containing this sense. If, however, a query word was used in one of it less common senses, expansion might be useful in ensuring that documents containing that sense was retrieved. Using this strategy of only expanding the less common senses of query words on the TREC queries 251–300 resulted in 4 queries being improved, 36 unchanged, and 5 degraded. Information on the frequency of occurrence of word senses was gained from WordNet and not from the collection the experiment was conducted on. The increased number of unchanged queries is not surprising given that fewer expansions took place.

3.1.2 Conclusions

The strategy of targeting query words using a less common sense may be a promising strategy, though obviously one that requires much improvement before it can be employed in any retrieval system. It has not yet been tested on the TREC-6 queries 301–350 and this is one of the future aims of this work.

3.2 Short vs long: small ones are more juicy?

As was stated in the introduction to this section, the results from the query expansion experiments were over shadowed somewhat by the results of the **glair64** submission showing that retrievals based on the description part of TREC queries were worse than retrievals based on the title sections. Contrary to expectations, it would appear that the compact queries of the title field are in general better than the more verbose queries of the description field.

3.2.1 Brief discussion

In this section a brief discussion of the possible reasons for these results are presented along with speculation on possible changes to query design in future TRECs.

Are long queries cursed? There is a well known result in retrieval research showing, in the context of relevance feedback at least, that there is an optimum size of query for producing the best retrieval effectiveness. This effect, sometimes called the 'curse of dimensionality'[12], has been shown to exist on a number of retrieval systems [2, 8, 3] including SIRE (the retrieval system employed in these experiments). Therefore, one explanation for the drop in effectiveness found in the **glair64** result could be due to this curse. Indeed, it does appear to be a factor. Figure 2 shows a scatter plot of average precision against query length for the 50 queries of TREC-6 (301–350), showing that at longer query lengths, average precision is generally lower. This trend, however, is not strong and other explanations should be examined before entirely blaming the result on the curse.

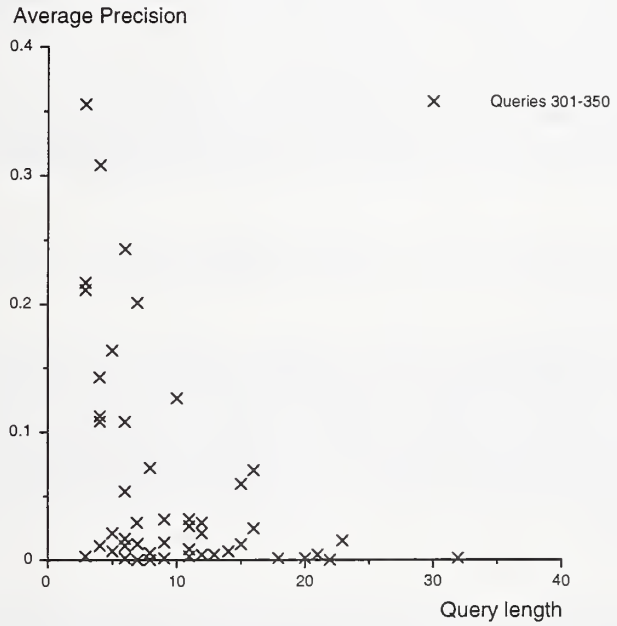


Figure 2: Scatter plot of average precision versus query length

Are the descriptions any good as queries? As can be seen in the two example queries in Figure 1, the description fields are written to be explana-

tions of information need intended for human consumption. From the point of view of a retrieval system, they contain seemingly useless phrases such as 'document will discuss' (phrases that seasoned TREC participants have in their stop lists) and sometimes clarifications that of information need that would be hard for a retrieval system to detect. Unless a retrieval system can parse the natural language of a description field, such subtleties will be lost. With this in mind, it is questionable if comparisons between the title and description sections are entirely fair as the two fields were not created for the same purpose. Indeed, there are a few queries in this year's TREC where one sees the title and description being used in a complimentary manner. For example query 349 requesting documents on the processes of living cells: the description contains rather general and ambiguous words, where as the title field is the single word 'metabolism' (rather like a question and accompanying answer). The very short version of this query produces good retrieval, but the longer version (minus this highly descriptive word) performs much worse. Like the previous explanation, it is not suggested that this difference between the description and the title fields is the sole reason for the drop in effectiveness on the longer queries, but it would appear to be a factor.

In order to eliminate it, it might be necessary to alter the guidelines for generating the description field possibly making it less of an naturally expressed request for information, more a simple list of words. In addition, it would be necessary to ensure that the title and description fields are kept independent of each other to avoid the complimentary type of query shown in Figure 1.

4 The Natural Language Track

We have developed a document retrieval model that uses noun phrases and single word terms for indexing and the retrieval processes [11]. The model is based on the Dempster - Shafer (D-S) theory of evidence [10] which is a generalisation of the Bayesian approach. The experiments were carried out on the 'B' collection.

4.1 Brief overview of the Dempster-Shafer theory

The D-S theory is a theory of uncertainty that assigns *belief* to propositions. A particular characteristic of the theory is that the belief of a proposition, x, does not necessarily imply that the belief associated to the negation of the proposition is $1 - x$ (as happens in probability theory). In the absence of

any other evidence to support the negation of the proposition, the remaining belief is assigned to the entire proposition set, and represents the *overall uncertainty* or *uncommitted belief*. The full understanding of the D-S theory is not the purpose of this paper. We only give the necessary information for the understanding of the document retrieval model developed.

The D-S theory uses a number in the range $[0, 1]$ to assign *exact* beliefs to mutually exclusive propositions of a *frame of discernment* Ω. The assignment is represented by a *basic probability assignment* usually denoted by m:

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{p \in \Omega} m(p) = 1$$

The belief values assigned must always sum to one. A belief assigned to Ω itself represents the uncommitted belief.

A fundamental function in the D-S framework is the belief function. The function calculates the total belief $\text{Bel}(p)$ committed to the proposition p, from the available evidence (as expressed by the basic probability assignment):

$$\text{Bel}(p) = \sum_{q \to p} m(q)$$

In contrast to the $m(p)$, which calculates the exact belief to p, $Bel(p)$ calculates the total belief committed to p.

4.2 Noun phrase extraction

We use a part of speech tagger module and a noun phrase extractor module for the extraction of noun phrases from the 'B' collection and TREC-6 queries 301–350. Tagging of all the text in document/query was performed followed by the extraction of several tag patterns considered to be noun phrases. Stop words were then deleted from noun phrases and the remaining words were stemmed using the Porter stemmer.

The Natural Language Processing modules used were designed and implemented at the Language Technology Group (LTG) of the Human Communication Research Centre (HCRC), University of Edinburgh. The tagger is a state-of-the-art tagger and is a resource used in the Knowledge Acquisition Workbench [4], currently under development. The tagger achieves 96-98% accuracy if all the words in the text are found in the taggers lexicon, and 88-92% if unknown words appear in the text.

4.3 Indexing and retrieval

4.3.1 Document indexing

Noun phrases extracted from documents were combined with single terms for the formation of a *frame of discernment* for the 'B' collection. For all the single terms of the document collection, all the 2^S boolean combinational elements were generated using the terms (S being their number), the negations ($\neg$) of these terms and the boolean conjunction ($\wedge$). These boolean elements represented the basic propositions of the constructed frame.

Suppose that a document collection contains only the two single terms "*red*" and "*wine*". We obtain the following four (basic) propositions in the frame Ω:

p_0	$\neg red \wedge \neg wine$
p_1	$\neg red \wedge wine$
p_2	$red \wedge \neg wine$
p_3	$red \wedge wine$

Any valid combination of the above four propositions (e.g., $p_1 \vee p_2$) is also a proposition of the frame Ω.

A basic probability assignment was associated with each document D_i. Its values were derived from the document frequency characteristics. The general weighting formula used in the first two runs (**Gla6DS1**, **Gla6DS2**) was:

$$m_i(p_j) = \begin{cases} \frac{\text{FREQ}_i(x_j)}{\text{TOTFREQ}_i} \cdot \log_N \frac{N}{n(x_j)} & j \neq 0 \text{ and } x_j \text{ is a term} \\ \frac{\text{FREQ}_i(x_j)}{\text{TOTFREQ}_i} \cdot \min\limits_{w \in x_j} \left\{ \log_N \frac{N}{n(w)} \right\} & j \in 0 \text{ and } x_j \text{ is a noun phrase} \\ 1 - \sum\limits_{x_k \in D_i} m_i(x_k) & p_j = \Omega \\ 0 & j = 0 \end{cases}$$

where:

1. p_j is the disjunction of (basic) propositions in the frame for which is constructed upon the single term or noun phrase x_j where x_j holds true. $p_0 = \perp$ so $m_i(p_0) = 0$. $m_i(\Omega) = m_i(\top)$ represents the uncommitted

belief of document D_i (Ω can be viewed as the disjunction of all the basic propositions (except $\perp$), that is the true proposition $\top$.

2. $\text{FREQ}_i(x_j)$ is the number of occurrences of x_j in document D_i.
3. $\text{TOTFREQ}_i = \sum_{x_k \in D_i} \text{FREQ}_i(x_k)$ is the number of total occurrences in document D_i.
4. $n(t_j)$ is the number of the documents in the collection that contain the term x_j.
5. $w \in x_j$ are all the single words in the noun phrase x_j.
6. $\log_N\left(\frac{N}{n(x_j)}\right)$ is the inverted document frequency (IDF) weight of the term x_j. We used the logarithm with base N so the IDF is in the interval $[0, 1]$.

The weighting schema used is version of the classic TF-IDF using normalised TF and normalised IDF. The TF factor is normalised with the length of the document (TOTFREQ_i) and the IDF factor is normalised with the logarithm of N. The D-S restriction for total belief being always equal to one motivated the normalised TF and IDF factors. The IDF value of noun phrases is always equal to the minimum IDF value of the single terms that constitute the noun phrase.

For the third run (**Gla6DS3**) the TF factor used is different for single terms. For each single term appearing in a noun phrase the frequency assigned to it is only the number of its occurrences in the document as a stand alone term (without counting its occurrences when it appears in a noun phrase).

4.3.2 Queries and Retrieval

The queries used in the three runs fall in these two categories:

Single term queries: Only single terms are used. This category was used in the first (**Gla6DS1**) and the third run (**Gla6DS3**).

Noun phrase queries: The noun phrases are extracted from queries were considered. The single terms that appear only in a noun phrase and not as stand alone single terms in a query, are used in the query only as part of the extracted noun phrase. This category of queries was used in the second run (**Gla6DS3**).

Queries are mapped onto the frame of discernment as a proposition:

$$Q = \bigvee_{p_k \in \text{query}} p_k$$

p_k are the propositions for terms x_k as defined in the document representation. The disjunction ($\vee$) is used since it is difficult to derive from a natural language query whether a user wants to find documents about "*red wine*" or documents about "*red*" or "*wine*" unless the former is found as a noun phrase in the query. If the term x_k can not be expressed as a proposition in the frame Ω then p_k is assigned the empty proposition $\perp$.

For measuring relevance of a query to a document the belief function of the D-S theory was used. The relevance of a document to a query is formulated as:

$$\text{Bel}_i(Q) = \sum_{p \to Q} m_i(p)$$

In documents where the belief value is zero there is no relevance of the document to the query. None of its indexing proposition implies the query proposition. For a document collection, all the estimated relevant documents to the query ($\text{Bel}_i(Q) > 0$) can be ranked using the belief value of each document for ranking. For example, a query with only the word "*wine*" will have belief value equal to the basic probability assigned to the propositions built upon the word "*wine*" (these are the propositions p_1 and p_3 in the table). A query with the noun phrase "*red wine*" will have belief value equal to the basic probability assigned to the propositions derived from the two words "*wine*" and "*red*" (this is the proposition p_3 in the table).

4.4 Results

The results obtained cannot be considered successful. Though the theoretical framework supporting the model is sound, the application of the proposed basic probability assignments and the belief function seems to lower precision when belief is given to noun phrases.

The main reason is that words with low IDF values are also existent in many noun phrases. For example, in the 'B' collection, the word "*account*" is a very frequent term. When it appears in noun phrases the belief value of the stand alone word increases. If a query requests for "*swiss account*" (interpreted as as a disjunction), a document containing the noun phrase "*current account*"

three times will be retrieved with high belief even though the word "*swiss*" is not contained in the document. This happens when the single word query approach is used (runs **Gla6DS1** and **Gla6DS3**).

A method for solving the above problems is to use the noun phrase queries (run **GlaDS2**). Unfortunately, this query approach retrieves only documents containing the noun phrase of the query. In the previous example the noun phrase "*current account*" will retrieve documents containing it but, it will not retrieve documents that have only the words "*swiss*" or "*account*" which are relevant to the query (though they do not contain the noun phrase "*swiss account*".

In brief, the main problem of the belief function as used in this model falls into two cases:

1. If single word queries are used it increases the belief of frequently unwanted terms in irrelevant documents, thus lowering dramatically precision.
2. If noun phrase queries are used the belief function is very specific in retrieval, and recall gets strongly affected.

Another major problem is the use of document length normalisation to the basic probability assignment which misleads the retrieval of short documents.

5 The Spoken Document Retrieval Track

5.1 The Abbot Speech Recognition System

The speech recognition system we used for the participation to the SDR track was kindly made available to us by the Speech and Hearing Research Group of the Department of Computing Science of the University of Sheffield track. We did not have to perform any speech recognition on the speech data, since we were given the transcripts by the Sheffield group. Nevertheless, we fell obliged to give a few details about the speech recognition system they used, referring back to their article at TREC-6 for more. The system they used is Abbot.

Abbot is a speaker independent continuous speech recognition system developed by the Connectionist Speech Group at Cambridge University and now jointly supported by Cambridge and Sheffield Universities with commercialisation by SoftSound.

The Abbot system grew out of a PhD on recurrent neural networks at the University of Cambridge. It was further developed under the ESPRIT project "Auditory Connectionist Techniques for Speech" and then the ESPRIT project "WERNICKE: A Neural Network Based, Speaker Independent, Large Vocabulary, Continuous Speech Recognition System". Currently further development is being funded by the Framework 4 projects "SPRACH: Speech Recognition algorithms for connectionist hybrids" and "THISL: Thematic Indexing of Spoken Language".

The system is designed to recognise British English and American English clearly spoken in a quiet acoustic environment. The system is based on a model that is a combination of a connectionist and a Hidden Markov model [6].

5.2 Experimenting Probabilistic Retrieval of Spoken Documents

In this section we report a brief account of the strategies we used for the two runs for the SDR track. A more detailed account of these techniques is reported in [1].

The PFT weighting schema

One of the characteristics of the data we had available from the Abbot speech recognition system is the uncertainty associated to each word recognised by Abbot. The following is an example of part of a srt file produced by Abbot.

```
<Episode Filename=a960521.sph Program="ABC_Nightline"
Scribe="obert_markoff" Dat e="960521:2330" Version=4 Version_Date=961011>
.
.
.
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960521.1" >
<Word S_time=1.76 E_time=2 Prob=-1.873> IT'S </Word>
<Word S_time=2 E_time=2.048 Prob=-0.9346> A </Word>
<Word S_time=2.048 E_time=2.656 Prob=2.025> QUESTION </Word>
<Word S_time=2.656 E_time=2.832 Prob=-0.6394> THAT </Word>
<Word S_time=2.832 E_time=2.992 Prob=-0.3682> WILL </Word>
<Word S_time=2.992 E_time=3.36 Prob=1.188> MAKE </Word>
<Word S_time=3.408 E_time=3.488 Prob=-0.9622> A </Word>
<Word S_time=3.488 E_time=3.872 Prob=2.335> LOT </Word>
```

```
<Word S_time=3.872 E_time=3.984 Prob=0.4647> OF </Word>
<Word S_time=3.984 E_time=4.672 Prob=5.322> AMERICANS </Word>
<Word S_time=4.672 E_time=4.864 Prob=-0.4521> THINK </Word>
<Word S_time=6.882 E_time=6.994 Prob=-2.392> TO </Word>
<Word S_time=6.994 E_time=7.234 Prob=-1.807> HAVE </Word>
<Word S_time=7.234 E_time=7.346 Prob=-3.124> TO </Word>
<Word S_time=7.91 E_time=8.086 Prob=-0.2239> YOU </Word>
<Word S_time=8.086 E_time=8.294 Prob=0.1139> SAY </Word>
<Word S_time=8.294 E_time=8.454 Prob=-2.961> TO </Word>
<Word S_time=8.454 E_time=8.95 Prob=-3.794> ONE </Word>
.
.
.
</Section >
```

These measures of uncertainty are incorrectly called probabilities, as an explanation of the way they are computed will clarify:

1. For a given time segment, the neural network at the heart of Abbot provides a set of posterior probabilities for each phoneme. These are the "acoustic probabilities".
2. To facilitate the decoding, the acoustic probabilities are converted into scaled likelihoods by dividing by the prior probability of the phoneme.
3. During decoding, a search is performed using the acoustic probabilities and the language model to find the most likely sequence of words for that utterance.
4. As each word is defined as a sequence of phonemes, the score for that word is obtained by summing the scores of the individual phones which constitute that word. (Summing because Abbot works with log probabilities).

Although they are not probabilities, we can still consider them as weights expressing the confidence given by Abbot in the correct recognition of words. This gave us the idea of combine these weights with the probabilistic model underlying SIRE.

The probabilistic model used by SIRE assigned to every index term extracted from the text of a document a weight that is a combination of two different discrimination measures: the IDF and the TF. The IDF of a term is a collection wide weight, since it is calculated taking into account the distribution of the term inside the whole collection. The TF of a term is instead a document wide weight, since it is calculated taking into account the distribution of a

term within a document. The TF is of particular interest in our discussion. The TF of a term is usually calculated as a normalised sum of the number of occurrences of that term in the document. If the occurrence of a term is a binary event, then:

$$occ.(x_j) = \begin{cases} 1 & \text{if } x_j \text{ occurs in } d_i \\ 0 & \text{otherwise} \end{cases}$$

Therefore, in its simplest definition, the frequency of occurrence of a term is defined as follows:

$$freq_i(x_j) = \sum_{d_i} occ.(x_j)$$

We decided to use the probabilities Abbot assigns to words as a way of devising a more general definition of occurrence. We decided to use the following definition of occurrence:

$$occ'.(x_j) = \begin{cases} Prob(x_j) & \text{if } x_j \text{ occurs in } d_i \\ 0 & \text{otherwise} \end{cases}$$

Therefore the frequency of occurrence of a term is now defined as:

$$freq_i(x_j) = \sum_{d_i} Prob(x_j)$$

This definition of frequency is the one used to redefine TF as follows:

$$PTF_{ij} = freq_i(x_j)$$

We called *PFT* (Probabilistic Term Frequency) this new definition of TF.

The above definition is quite intuitive. While TF measures the importance of a term in the context of a document as a function of the number of occurrences of the term, PTF weights the number of occurrences of a term with

the confidence assigned every time to the recognition of the occurrence of the term. In fact, it is intuitive that the PTF of a term should be higher in the case the term being recognised as present in the document with high confidence values, that in the case of being recognised with low confidence values. In the latter case, in some instances, the term may have been mistaken for another term and may not even be present in the document.

In some of the experiments that follow we tried to see if a PTF-IDF weighting schema gives better performance than the classical TF-IDF. The actual formula for the PTF used in these experiments is, for reasons that we will not discuss here, the following:

$$PTF_{ij} = K + (1 - K)\frac{freq_i(x_j)}{maxfreq_i}$$

Generating a weighting schema by merging different transcriptions

In the previous section we have taken advantage of a particular feature of the transcription we had available, the probabilities assigned by Abbot to words in the transcription. We used these probabilities to generate a new weighting schema for the words in the transcription. However, a few questions that we posed ourself were: are these probabilities reliable? Is there any other strategy that we could use to generate confidence (or uncertainty) values to assign to recognised words?

Another, perhaps naive, strategy that we decided to test was again due to our particular situation. We had availability of two different speech recognition transcript for the same speech data. A first analysis of the two transcripts shows large differences in recognition. Here is a short example:

BSRT (NIST/IBM recogniser) :

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960523.1" >

I will talk about blacks and winds we eventually go wrong a
of the tough question who he hid ...

</Section>
```

Abbot (Sheffield recogniser) :

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960523.1" >

we talked about blanks and whites we eventually get around
```

```
to the tough question his own unions say well ....

</Section>
```

DTT (Actual transcript) :

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960523.1" >

when we talk about blacks and whites we eventually get around
to the tough question some of you are ...

</Section>
```

It is easy to spot the errors made by the two speech recognition systems. One interesting fact is that there are many cases of words correctly recognised by one system and wrongly by the other. For example, the word "blacks" has been correctly recognised by BSRT and wrongly by Abbot, while the word "white" has been correctly recognised by Abbot and wrongly by BSRT. If one of these two words would have been used in a query, the IR system could not avoid retrieving only the document in which the word has been recognised correctly.

This suggested merging the two speech recognition transcripts. In this case the correct recognition of one system could compensate for the wrong ones of the other system. Moreover, using the classical TF-IDF weighting schema, if a word has been correctly recognised by both systems, then it will have a larger frequency of occurrences and this will increase its weight in the context of the document. On the other hand, a word that has been wrongly recognised by one of the speech recognition systems will have a small frequency of occurrence (unless it has been consistently recognised wrongly, a case that we suppose does not happen frequently) and therefore a lower weight in the context of the document. We called *Merged* this weighting schema.

5.3 Results

We will not discuss the figures returned from TREC in detail in this paper. We will just note that:

- the R1 run (**gla6R1**, using hand transcripts) is right on the median value;
- the B1 run (**gla6B1**, NIST/IBM data) is slightly above the median value;

- the S1 run (**gla6S1**, using the PTF strategy with Abbot data) is below the median value, clearly, if the PTF weighting scheme is to be of any use, it requires further work;
- the S2 run (**gla6S2**, using a merged NIST/Abbot collection) is above the median value and better than both the B1 run and the S1 run. In fact, under some of the evaluation measures listed in the results file (particularly the mean reciprocal) the S2 run is almost as good as the R1 run: the manual transcripts! In all the tests using merging, we found it to be always better than retrieval on the individual collections and we feel this provides some evidence towards regarding merging transcripts as a consistently good strategy in retrieval of spoken documents.

5.3.1 Conclusions and future works on SDR

This was our first experience in dealing with retrieval of spoken documents and we are pleased with the results of the initial efforts. Cross comparisons between groups with their alternate IR strategies and different recognisers is not easy. Our impression of the trend of results, however, is that no amount of clever retrieval strategies will compensate for a poorly recognised transcript. We certainly feel that our relative success in retrieving spoken documents has much to do with the quality of transcript generated by the Abbott System of Sheffield University.

6 Conclusions

To conclude, our participation to TREC-6 was a very interesting one and useful one in all three the tracks we took part in. The results achieved, that we only briefly reported in this paper but that are summarised at the end of this proceedings, encourage us to pursue our future participation for next TREC at least in the short queries and in the SDR tracks.

References

[1] F. Crestani, and M. Sanderson. Retrieval of Spoken Documents: First Experiences. Departmental Research Report TR-1997-34, Department of Computing Science, University of Glasgow, UK, October 1997.

[2] D. Harman. Relevance feedback revisited. In *Proceedings of ACM SIGIR*, pages 1–10, Copenhagen, Danmark, June 1992.

[3] M. Magennis and C.J. van Rijsbergen. The potential and actual effectiveness of interactive query expansion. In *Proceedings of ACM SIGIR*, pages 324–332, Philadelphia, PA, USA, July 1997.

[4] A. Mikheev and S. Finch. A workbench for finding structure in texts. In *Proceedings of the Applied Natural Language Processing (ANLP-97), Washington D.C.*, April 1997.

[5] G. A. Miller. A lexical database for english. *Communication of the ACM*, 38(11):39–41, 1995.

[6] T. Robinson and M. Hochberg and S. Renals. The use of recurrent networks in continuous speech recognition. In C. H. Lee and K. K. Paliwal and F. K. Soong, editors, *Automatic Speech and Speaker Recognition – Advanced Topics*, pages 233–258. Kluwer Academic Publishers, 1996.

[7] M. Sanderson. System for information retrieval experiments (SIRE). Unpublished paper, November 1996.

[8] M. Sanderson. *Word Sense Disambiguation and Information Retrieval.* PhD Thesis, Department of Computing Science, University of Glasgow, Glasgow, Scotland, UK, 1996.

[9] M. Sanderson and R. Wilkinson. Guidelines for generating very short TREC queries, 1997. `http://www.dcs.gla.ac.uk/~sanderso/guidelines.html`.

[10] G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, NJ, 1976.

[11] Marcos Theophylactou. Document Retrieval using Natural Language Processing and the Dempster - Shafer Theory of Evidence. Master's thesis, University of Glasgow, Department of Computing Science, September 1997.

[12] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, second edition, 1979.

[13] E.M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of ACM SIGIR*, pages 61–69, Dublin, Ireland, July 1994.

Document Translation for Cross-Language Text Retrieval at the University of Maryland *

Douglas W. Oard and Paul Hackett
Digital Library Research Group
College of Library and Information Services
University of Maryland, College Park, MD 20742
{oard,pghtwoz}@glue.umd.edu

Abstract

The University of Maryland participated in three TREC-6 tasks: ad hoc retrieval, cross-language retrieval, and spoken document retrieval. The principal focus of the work was evaluation of a cross-language text retrieval technique based on fully automatic machine translation. The results show that approaches based on document translation can be approximately as effective as approaches based on query translation, but that additional work will be needed to develop a solid basis for choosing between the two in specific applications. Ad hoc and spoken document retrieval results are also presented.

1 Introduction

The principal goal of the University of Maryland's participation in the Sixth Text REtrieval Conference (TREC-6) was to evaluate the performance of a document translation strategy for Cross- Language Information Retrieval (CLIR). The Logos machine translation system[1] was used in a fully automatic mode for both document and query translation, and Inquery release 3.1p1 from the University of Massachusetts[2] was used for all runs. We participated in the Ad Hoc task as well in order to establish a baseline for the performance of this version of Inquery, and we also used Inquery for Quasi-Spoken Document Retrieval (QSDR) track runs in preparation for future work on speech-based information retrieval. No manual processing was done, and all of our runs were submitted in the automatic category.

2 Cross-Language Information Retrieval

Query translation has emerged as the most popular technique for CLIR, typically achieving between 50% and 75% of the retrieval effectiveness that is reported for comparable monolingual techniques when coupled with simple linguistic processing such as part-of-speech tagging or phrase indexing [4]. Query translation strategies are relatively efficient when short queries are presented, but a lack of adequate linguistic context in queries containing only a few words may limit the ability of systems to select the most appropriate translations for the query terms. Machine translation systems seek to exploit contextual clues in full-length documents to produce the best possible translations, and it is an open question whether a retrieval system based on automatic machine translation of each document can outperform query translation. We have thus sought to determine whether the additional effort required to translate every document would produce better retrieval effectiveness than query translation for the TREC-6 CLIR track.

The Logos machine translation system that we used for our experiments is a commercial product that is designed to assist human translators by automatically preparing fairly good translations of individual

*This work has been supported in part by DARPA contract N6600197C8540 and the Logos Corporation.
[1]Logos Corporation, 111 Howard Boulevard, Suite 214, Mount Arlington, NJ 07856 USA
[2]Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA 01003

Technique	Title	Short	Long
Unstopped Monolingual	0.2480	0.1219	0.2396
Document Translation	0.1761	0.1829	0.2171
Query Translation	0.1668	0.1492	0.1561

Table 1: Non-interpolated average precision for the SDA/NZZ collection, averaged over 21 topics.

documents. The system is typically used by translation bureaus and other organizations as the first stage of a machine-assisted translation process, and we have previously used it for cross-language routing experiments [3]. The Logos system includes extensive facilities for adding domain-specific technical terminology and new linguistic constructs, but for TREC-6 we used only the machine readable dictionaries and semantic rules that are delivered as standard components of the product. The entire SDA and NZZ collections were translated from German into English, and only format-related preprocessing and postprocessing was performed. A brief description of the translation process is contained in Appendix A. The translated documents are available to TREC participants through the NIST FTP site, and the README file with those documents contains sufficient detail to reproduce the translation runs.

We used four SPARC 20 workstations and a fifth workstation that was upgraded from a SPARC 5 to a SPARC Ultra 1 after about three quarters of the documents had been translated. All of the workstations were shared with other users. Translation of the 48 months of news stories contained in the SDA and NZZ collections using these machines required approximately 2 months. About half of the CPU time was required to perform the translations themselves, the remainder being shared with other users of the same machines or lost due to operator- or system-induced problems. Even with these problems, this works out to a single-machine translation rate that is at least 5 times faster than the rate at which the news articles were originally generated.

Once all of the documents had been translated into English, a single Inquery index was built for the union of the SDA and NZZ collections. Index construction required a two hours on a dedicated Sparc 20, and retrieval results for all 25 queries were typically computed in a few minutes (varying slightly with query length). Approximately 5% of the translations, almost entirely NZZ documents, were unavailable when the original index was constructed, but those translations have been subsequently completed and are included in the corrected runs presented here. Appendix C relates these corrected runs to the official results scored by NIST.

Table 1 summarizes the non-interpolated average precision results for three retrieval approaches, averaged over the 21 topics for which relevant documents are known in the SDA/NZZ collection, and Figure 1 shows recall-precision graphs for the same data.[3] Three query lengths were used: only words appearing in the title field ("title"), only words appearing in the desc field ("short"), and all words appearing in the topic description except SGML markup ("long"). As Table 1 shows, short queries were not as good as titles alone, and a query-by-query analysis revealed greater variation across topics for short queries as well. We used words from the title field in both our "title" and "long" queries, and it is possible that omitting those (usually very informative) words from our "short" queries offset any improvement that might otherwise have resulted from extending the length of the query. In Figure 1 and what follows we have chosen to focus on title and long queries since including short queries would likely contribute more to clutter than to clarity.

The monolingual retrieval results in Figure 1 provide a useful baseline for evaluating cross-language retrieval performance. In those runs we used the untranslated SDA/NZZ document collection and the German queries. We did not have a German stemmer available, but we did construct a small stopword list (see Appendix B). As Figure 2 shows, the use of that German stopword list adversely impacted long queries and had no impact on title queries, so we have presented only unstopped results when using German documents.

For the document translation runs we used the Logos translations of the SDA/NZZ documents into

[3]No relevant documents are known in the German SDA/NZZ collection for topic CL22, and relevance judgments are not available for topics CL03, CL15 and CL25.

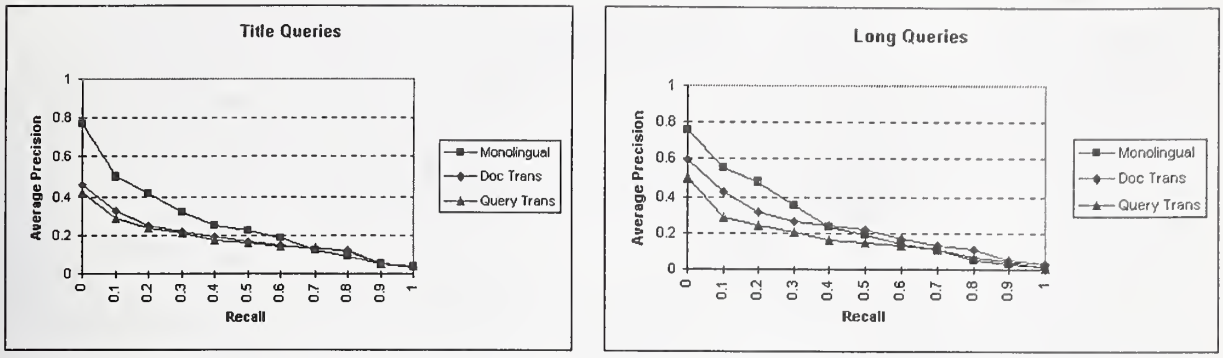
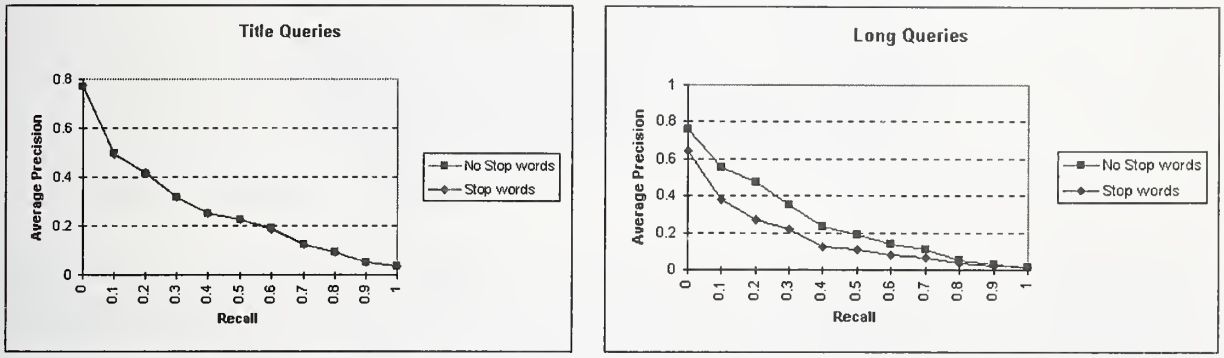


Figure 1: Comparison of retrieval approaches on the SDA/NZZ collection.



Title queries

Long queries

Figure 2: Recall-precision for monolingual retrieval on the SDA/NZZ collection with and without stopwords.

English and the English queries. Unlike the monolingual runs, both stemming and stopwords were used for the document translation runs. We used the Inquery "kstem" stemmer and Inquery's standard English stopword list. All other Inquery parameters were identical between the two sets of runs.

The SDA/NZZ query translation runs were made by using Logos to translate the English queries into German. The resulting queries were then used to retrieve untranslated SDA/NZZ documents. Again, Inquery was used without stemming or stopwords when processing German documents. Since Logos generates only a single "best guess" translation for any input, this approach differs in an important way from the more common approach based on cross-language query expansion. Cross-language query expansion techniques typically seek to replace each term in the query with every reasonable translation, including more than one possibility whenever unresolvable ambiguity is present [2]. By contrast, in the face of ambiguity Logos will simply choose whatever appears to be the best single translation.

Figure 1 shows that document translation and query translation perform about equally well on title queries, but that some advantage for document translation is apparent for long queries. Figure 3 depicts this result another way, showing the gain in uninterpolated average precision that results from using document translation rather than query translation on a query-by-query basis. Topic CL19 appears to account for much of the improvement in the long queries. It is difficult to draw strong conclusions from these results alone because the Logos "winner take all" approach to query translation has not been previously evaluated, but it does appear that document translation is performing at least as well as query translation and that both approaches are performing creditably, with results for title and long queries ranging between 67% and 90% of monolingual average precision on the SDA/NZZ collection.

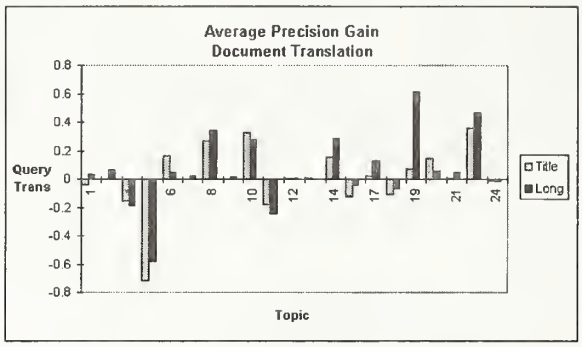


Figure 3: Relative advantage of document translation on the SDA/NZZ collection.

Technique	Title	Short	Long
Stopped Monolingual	0.3449	0.3121	0.3958
Query Translation	0.1928	0.1975	0.2455

Table 2: Non-interpolated average precision for the AP collection, averaged over 21 topics.

We did not try document translation on the CLIR track English AP collection, but we have obtained query translation and monolingual retrieval results for that collection using the untranslated AP documents, the "kstem" stemmer, and the standard Inquery stopword list. Table 2 and Figure 4 show those results. The monolingual results were obtained using English queries, while the query translation results were obtained with queries translated from German into English by Logos. Not surprisingly, a comparison of the results in Table 2 with those in Table 1 shows that retrieval effectiveness varies substantially across document collections, even when the same topics are used.

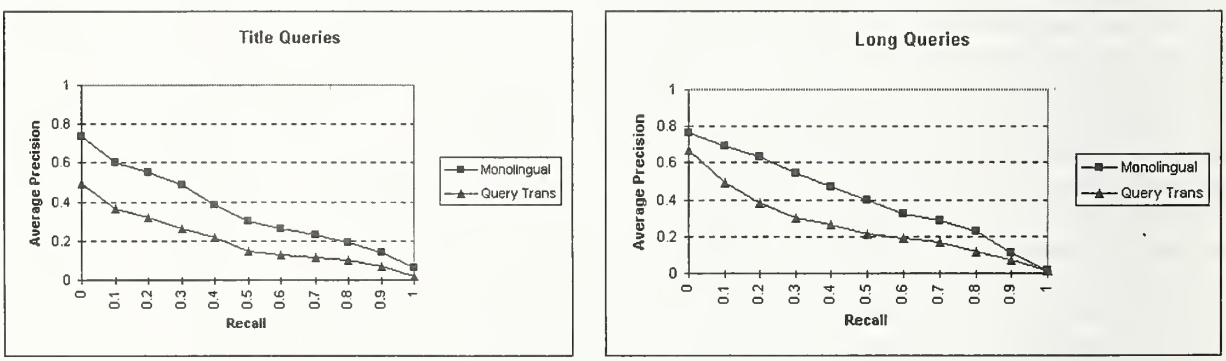


Figure 4: Query translation and monolingual retrieval results for the AP collection.

3 Ad Hoc Task

We used our participation in the ad hoc retrieval task to characterize the performance of our Inquery configuration in comparison with a broad range of participating systems. We submitted a single category A run with short queries based solely on the description field of each topic. Except for some content-neutral

preprocessing to handle differing SGML markup, we used the same Inquery configuration for the ad hoc task that we used for our cross-language runs. The resulting non-interpolated average precision, averaged over 50 topics, was 0.1460. As Figure 5 shows, we achieved at or above median average precision for 33 of the 50 topics. It is difficult to draw strong inferences from this, however, given the general dissatisfaction with the performance of short queries on the ad hoc task this year. This was our first Category A submission, and we learned the usual lessons about the consequences of initially allocating far too little time and not quite enough disk space to the effort. We had no prior experience with Inquery and we estimate our overall effort to produce these results at 1 person-month. Based on installation effort and retrieval effectiveness, our assessment is that Inquery offers a practical alternative to the SMART version 11.0 system that we used in TREC-5 for modular cross-language retrieval experiments in which the translation and retrieval components are loosely coupled. We have not yet explored the Inquery API in sufficient detail to assess whether it will be practical to use Inquery to investigate more tightly coupled approaches in which unresolvable translation ambiguity must be preserved.

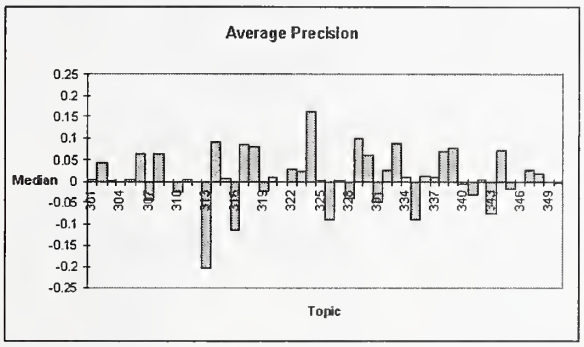


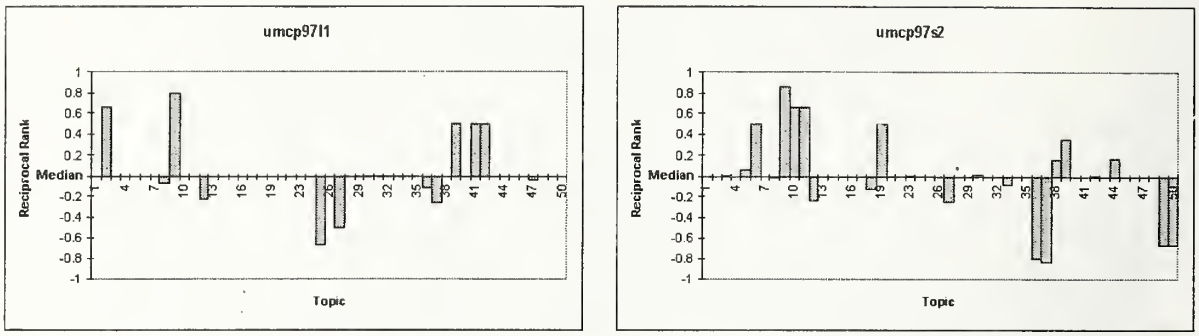
Figure 5: Monolingual retrieval for the ad hoc task.

4 Spoken Document Retrieval

We have recently initiated a project to investigate user interface design for information retrieval systems that provide access to large collections of recorded speech [5], and the Spoken Document Retrieval (SDR) track offered our first opportunity to gain experience with content-based retrieval using speech recognition output. We used Inquery to produce both a reference run from the transcripts and a QSDR run on the baseline recognizer output. Except for format-specific preprocessing, we made no other changes to our Inquery configuration for those runs. Figure 6 shows relative reciprocal ranks for our reference transcript and baseline recognizer runs, compared with the median reciprocal rank for each case. As Figure 7 illustrates, retrieval effectiveness declined substantially on about one quarter of the topics when the baseline recognizer output was substituted for the manually prepared transcripts.

5 Future Work

We are interested in exploring whether further improvements in cross-language retrieval effectiveness can be achieved by using the sort of linguistic analysis found in modern machine translation systems, but retaining any unresolvable ambiguity in a manner that can be effectively used by a text retrieval system. We are considering two approaches to this problem, one based on the extraction of intermediate representations from an existing machine translation system, and a second based on incorporation of more sophisticated linguistic representations into the retrieval system itself. This later approach has produced disappointing results in monolingual retrieval applications (c.f., [6]), but we believe that the presence of translation ambiguity in cross-language retrieval transforms the problem into one for which more sophisticated representations may



Reference transcript Baseline recognizer

Figure 6: Speech Data Retrieval results — reciprocal rank vs. median reciprocal rank by query.

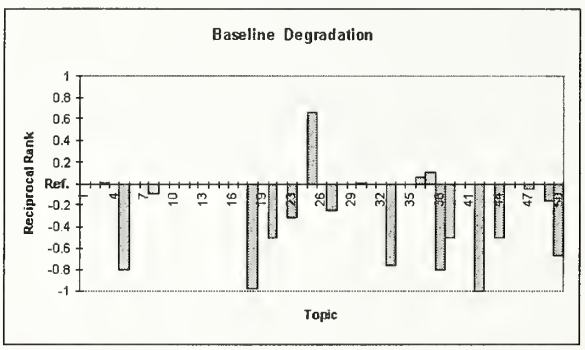


Figure 7: Degradation in reciprocal rank due to recognition errors.

be useful. Both of the approaches that we are considering should be able to exploit the linguistic context that is present in either documents or long queries, so both lead us in the direction of further experiments on cross-language retrieval based on document translation.

6 Conclusions

We have shown that document translation is a practical approach for cross-language text retrieval on moderately large collections, and we have observed some indications that document translation may ultimately be more effective than query translation for some applications. It appears that the CLIR test collection that has been developed at TREC-6 will be extremely useful for further investigation of these issues, and that is undoubtedly the most important legacy of this work. By providing a standard benchmark for evaluating the performance of competing approaches, the CLIR track has provided a sound basis for further advances in cross-language information retrieval.

Acknowledgments

The authors are grateful to Bonnie Dorr for providing facilities, resources and advice, to Scott Bennett and Harriet Leventhal for their assistance with the Logos translation system, to the University of Massachusetts for the use of Inquery, and to James Allan for help with Inquery configuration.

Appendices

A Document Translation Process

The translations were performed completely automatically using release 7.8.1 (or, for some NZZ documents, release 7.8.2) of the Logos machine translation system. System parameters were selected to use all available dictionaries and to maintain the imperative form where possible, but no new dictionaries were created for this purpose. The output was converted to the ISO 8859-1 (Latin-1) character set. Words that were not recognized by the Logos machine translation system were maintained in the original German, but characters with diacritical marks were mapped to the corresponding unmarked character.

In the SDA collection, only the LD, TI, TB, and TX fields were translated and indexed. In the LD field, the portion of the first line preceding the first ")" character was not translated. A total of 55 SDA documents failed to translate at all due to system errors. Those documents were removed from the translated collection but the corresponding untranslated documents were retained for the monolingual and query translation runs.

In the NZZ collection, the INDENT_TEXT, FOOTNOTE, TEXT, MAIN_TITLE, MAIN_TITLE_1, KURSIV_TITLE, KURSIV_TITLE_1, KURSIV_TITLE_2, LEAD, LINE_TITLE, LEGEND, MAGAZINE_TITLE, HEAD_TITLE, HEAD_TITLE_1, POETRY_TEXT, COLUMN_TITLE, SIDEHEAD_TEXT, FOOT_TITLE, FOOT_TITLE_1, FOOT_TITLE_2, INTRO_PARA, QUOTATION, SECTION_TITLE, and SECTION_TITLE_1 fields were translated and indexed. A total of 174 NZZ documents failed to translate due to system errors. Those documents were removed from the translated collection that was used for the document translation runs but the corresponding untranslated documents were retained for the monolingual and query translation runs.

B German Stopword List

The German stopword list that we tried for monolingual German runs was constructed by manually selecting stopwords from the German lexicon described in [1]. Terms were selected from prepositions, other functional elements, complementizers, pronouns, and a few contractions and other words, and selections were made by the developer of the lexicon, a non-native speaker of German. The following list contains every word in our stopword list:

ab aber alle allen aller am an andere anderem anderen anderer anderes ans auf auf aufwaerts aus bei beim das dein dem den denn der des dich die diese diese diesem diesen dieser dieser dieses dir drei dreie dreien dreier du du ein ein eine einem einen einer eines einige einigen einiger er es es euch euer für heraus herein herunter hinaus hinein hinter hinunter ich ihm ihn ihnen ihr im in ins jede jedem jeden jeder jedes jemand jemand jene jenem jenen jener jenes keine keinem keinen keiner keines man mein mein mich mir mit nach neben niemand niemand ob ohne sein selbst sich sich sie sie sie so über um und uns uns unser unser unter unter verschiedene verschiedenen verschiedener viele vielen vieler von vor wann warum was wegen weil weil welche welchem welchen welcher welches wem wen wer wes wessen wie wieviele wievielem wievielen wievieler wievieles wir wo zehn zu zu zum zur zwei zweie zweien zweier

C Official TREC Runs

Translations for approximately one sixth of the NZZ documents (scattered throughout the year) were not available in time for the official TREC submission, so those documents were not present in the translated collection that was used for the document translation runs. Formatting errors in the construction of two long English queries also resulted in submission of one official run without any selections for those topics. The results presented above reflect the corrected runs. Table 3 shows the correspondence of those runs to the identifiers of the official TREC runs.

D CLIR Track Questionnaire

1. OVERALL APPROACH:

Identifier	Collection	Queries	Approach	Remarks
umcpxgg1	SDA/NZZ	Title	Stopped monolingual	
umcpxgg2	SDA/NZZ	Short	Stopped monolingual	
umcpxgg3	SDA/NZZ	Long	Stopped monolingual	
umcpxgg4	SDA/NZZ	Title	Unstopped monolingual	
umcpxgg5	SDA/NZZ	Short	Unstopped monolingual	
umcpxgg6	SDA/NZZ	Long	Unstopped monolingual	
umcpxeg1	SDA/NZZ	Title	Document translation	
umcpxeg2	SDA/NZZ	Short	Document translation	
umcpxeg3	SDA/NZZ	Long	Document translation	Added CL12 and CL17
none	SDA/NZZ	Title	Query translation	New run
none	SDA/NZZ	Short	Query translation	New run
none	SDA/NZZ	Long	Query translation	New run
none	AP	Title	Stopped monolingual	New run
none	AP	Short	Stopped monolingual	New run
none	AP	Long	Stopped monolingual	New run
umcpxge1	AP	Title	Query translation	
umcpxge2	AP	Short	Query translation	
umcpxge3	AP	Long	Query translation	

Table 3: Official TREC identifiers corresponding to the corrected runs.

```
  1.1 What basic approach do you take to cross-language retrieval?
      [X] Document Translation

  1.2 Were manual translations of the original NIST topics used as a
      starting point for any of your cross-language runs?
      [X] No

  1.3 Were the automatically translated (Logos MT) documents used
      for any of your cross-language runs?
      [X] Yes, umcpxeg1, umcpxeg2, umcpxeg3

  1.4 Were the automatically translated (Logos MT) topics used
      for any of your cross-language runs?
      [X] Yes, umcpxge1, umcpxge2, umcpxge3

2. MANUAL QUERY FORMULATION:  N/A

3. USE OF MANUALLY GENERATED DATA RESOURCES:

  3.1 What kind of manually generated data resources were used?
      [X] Part-of-speech Lists (for stopword list development)

  3.2 Were they generated with information retrieval in mind or were
      they taken from related fields?
      [X] Machine Translation

  3.3 Were they specifically tuned for the data being searched (i.e.,
      with special terminology) or general-purpose?
```

[X] General purpose

3.4 What amount of work was involved in adapting them for use in your information retrieval system.
[X] 15 minutes

3.5 Size: See Appendix B

3.6 Availability: The source of the original part of speech list is cited in paper, the stopword list is provided in Appendix B.

3. USE OF MANUALLY GENERATED DATA RESOURCES:

3.1 What kind of manually generated data resources were used?
[X] Other, Logos MT

3.2 Were they generated with information retrieval in mind or were they taken from related fields?
[X] Machine Translation

3.3 Were they specifically tuned for the data being searched (i.e., with special terminology) or general-purpose?
[X] General purpose

3.4 What amount of work was involved in adapting them for use in your information retrieval system.
[X] 1 week

3.5 Size
[X] Est. 40,000 word dictionary

3.6 Availability? - Please also provide sources/references!
[X] Commercial, cited in paper.

4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES: N/A

5. GENERAL

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?
[X] Easily replaceable

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?
[X] Yes, somewhat

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?
[X] Yes, somewhat

5.4 Are similar resources available for other languages than those used?

[X] Yes, analysis in German and English, generation in German, English, Italian, French, Spanish

References

[1] Bonnie J. Dorr. *Machine Translation: A View from the Lexicon.* MIT Press, Cambridge, MA, 1993.

[2] David A. Hull and Gregory Grefenstette. Experiments in multilingual information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996. http://www.xerox.fr/people/grenoble/hull/papers/sigir96.ps.

[3] Douglas W. Oard. Adaptive filtering of multilingual document streams. In *Fifth RIAO Conference on Computer Assisted Information Searching on the Internet*, June 1997. http://www.glue.umd.edu/~oard/research.html.

[4] Douglas W. Oard. Alternative approaches for cross-language text retrieval. In *AAAI Symposium on Cross-Language Text and Speech Retrieval.* American Association for Artificial Intelligence, March 1997. http://www.glue.umd.edu/~oard/research.html.

[5] Douglas W. Oard. Speech-based information retrieval for digital libraries. Technical Report CS-TR-3778, University of Maryland, College Park, March 1997. http://www.glue.umd.edu/~oard/research.html.

[6] Mark Sanderson. Word sense disambiguation and information retrieval. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 142–151. Springer-Verlag, July 1994. http://www.dcs.gla.ac.uk/ir/papers/Postscript/sanderson94b.ps.gz.

Between Terms and Words for European Language IR and Between Words and Bigrams for Chinese IR

Jian-Yun Nie

Département d'Informatique et Recherche opérationnelle,
Université de Montréal
C.P. 6128, succursale Centre-ville
Montreal, Quebec, H3C 3J7 Canada

nie@iro.umontreal.ca

Jean-Pierre Chevallet,
Marie-France Bruandet

Laboratoire CLIPS, IMAG
BP. 53X, 38041 Grenoble cedex, France

Jean-Pierre.Chevallet@imag.fr
Marie-France.Bruandet@imag.fr

Université de Montréal, together with the MRIM research group of the CLIPS laboratory in IMAG institute, participated in the Cross-Language Retrieval track in TREC6. Université de Montréal also participated in the Chinese track. In this paper, we describe our approaches used in our experiments. In the cross-language retrieval track, we compared word-based retrieval and term-based retrieval. In the Chinese track, the approaches using bigrams and words are compared.

1. Introduction

The principal goal of our participation in TREC6 is to compare the following two pairs of approaches:

For French and Cross-language retrieval:

- the classical approach based on words;
- our approach using terms from a terminological base, and automatically built terms.

For Chinese retrieval:

- the approach using bigrams;
- the approach using word segmentation.

This report describes our approaches and the experimental results.

2. French and Cross-Language Retrieval using terms

Classical IR systems operate on a word-basis. That is, both documents and queries are represented by a set of weighted words (keywords). This approach has been criticized in a number of studies on the two following points:

1. The content of a document (or a query) cannot be represented precisely by a set of isolated words.

2. Different keywords are assumed to be independent. In reality, they are not.

To solve the first problem, it is often suggested that compound terms, instead of single words, should be used. For example, when the compound term "expert system" is used to represent a document content, it is more precise than using "expert" and "system" separately. Many studies have been concerned with the problem of extracting compound terms from texts. Many of them are based on statistics of word co-occurrences and syntactic analysis of phrases.

However, whether such an approach can bring significant improvements to the system's performance is still an issue [3, 10].

To solve the second problem, it is generally suggested that one must incorporate in the query evaluation some form of inference using relationships between terms. The following approaches have been proposed to establish relationships between different terms:

- by considering term co-occurrences in the document collection [9], or
- by considering user relevance feedback [4, 5].

Terminological bases may provide solutions to both problems. A terminological base may contain a large number of compound terms set up by experts. They can be used to index documents and queries. In addition, in some bases, relationships are established between different terms. These relationships may be used to retrieve related documents.

However, a manually established terminological base may not cover correctly all the terms in a particular application domain. An automatic term building mechanism may be useful to complement a manual base. In our experiments, we tested the use of a manual terminological base both alone and in combination with an automatically built base, using the French document collection and French queries. Later on we also used the manual base for English to French retrieval.

2.1. Using manual terminological base

We used the "Banque de Terminologie du Québec" (BTQ) as our source of terms. The BTQ has been developed by the "Office de la Langue Française du Québec" (Office of French Language of Quebec). It contains over 500 000 files classified in about 160 different domains. Each file in the BTQ contains (among others) a term, its domains, related terms, and definition. All these elements are in both French and English. Below is part of the file 1000012 (file id.) in the BTQ:

```
1000012    11 (domain)
       1615 (pyrotechny)
1000012    13 (English term)
       delay electric blasting cap
1000012    17 (English definition)
       An electric blasting cap with a delay element between the priming
       and detonating composition to permit firing of explosive charges in
       sequence with but one application of the electric current.
1000012    41 (French term)
       détonateur électrique à retard
1000012    45 (French definition)
       Ces détonateurs permettent d'échelonner plusieurs explosions dans
       le temps en n'employant qu'un seul circuit électrique et une seule
       charge électrique.
1000012    48 (French synonyms)
       amorce électrique à retard
       détonateur électrique à retardement
```

The BTQ has a good coverage for a number of specialized domains (especially some

scientific domains). Our intention is to use the BTQ to do the following processes:

- Using the terms in the BTQ to extract terms in both documents and queries in order to create a additional representation more precise than words;

- Using the synonyms (and other related terms) to extend the user's queries;

- In the cross-language retrieval, English terms recognized in a query are translated into their corresponding French terms which are then used to retrieve French documents.

The extraction of the BTQ terms from texts may be seen as illustrated in Figure 1.

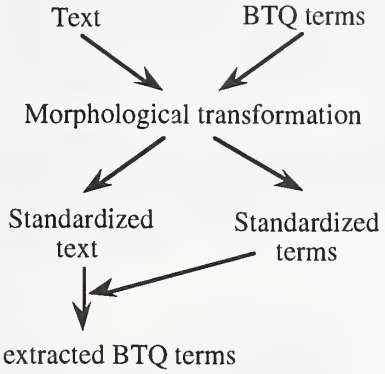


Figure 1. Process flow

The first step aims to transform the terms in the BTQ into a standard form with a stemming process. A French text is submitted to the same transformation. The stemming process removes 123 frequently used French word endings. For example:

- issement (élargissement)
- amment (indépendamment)
- able (traitable)
- ance (plaisance)

After the transformation, the text and the terms in the BTQ may be compared so that all the terms occurring in the text may be recognized and extracted. Below is an example to illustrate the result of the extraction.

Initial text

Description et schéma d'une cellule d'écoulement avec une électrode de carbone vitreux recouverte d'un film de Hg, utilisée pour cette méthode d'analyse. Etude de l'influence de la concentration en Hg2+ et de l'oxygène dissous sur la hauteur du signal lors du dosage de Pb et Cd. Observation d'une augmentation de la sensibilité avec le système en continu par rapport à un système en discontinu

Extracted terms:

Description, schéma, cellule, écoulement, électrode, électrode de carbone, carbone, vitreux, film, Hg, méthode, analyse, concentration, oxygène, oxygène dissous, dissous, hauteur, signal, dosage, Pb, Cd, Observation, augmentation, sensibilité, système, continu, rapport, système, discontinu

2.2. Automatic term building

This task was done by the IOTA system. The IOTA system was built in the middle of 1980s. It contains a component that builds a term base automatically from corpus [1]. This process is based on a syntactic analysis of French sentences, as well as the frequency of word co-occurrences in the corpus.

In order to build a term base, the following processes are performed: First, the texts in the corpus are analyzed so that a (sometimes more than one) syntactical category is attached to each word. This requires us to solve many ambiguities during the analysis. To do this, a precedence matrix is used which tells if a given category can follow or precede another category in French. In this way, many impossible sequences of syntactic categories may be removed. After the tagging, all the word groups that fit one of the pre-determined syntactic structures are identified as potential terms. Finally, a statistical analysis is used to determine a degree of potentiality for each group. Only highly potential groups are retained as terms and are put into the automatic term base. Below are some examples of the terms recognized by this process (together with their frequency in the corpus):

2394 conseil_fédéral	1026 grand_conseil	647 année_dernier
2025 premier_foi	1015 police_cantonale	601 croix_rouge
2017 million_de_franc	991 conseil_de_administration	596 annoncer_jeudi
1754 new_york	984 droit_de_homme	588 ronald_reagan
1672 premier_ministre	861 office_fédéral	578 également_été
1455 conseil_national	840 pouvoir_être	570 annoncer_mardi
1435 nations_unies	819 département_fédéral	568 anné_précédente
1385 affaire_étrangère	815 dernier_année	560 perez_de_cuellar
1326 chiffre_de_affaire	770 parti_communist	553 annoncer_mercredi
1243 million_de_dollar	725 source_officie	550 cour_de_conférence
1213 grande_bretagne	724 déclarer_m	548 nord_ouest
1164 affaires_étrangère	701 territoire_occuper	540 suisse_romand
1153 milliard_de_dollar	691 semaine_dernier	529 annoncer_vendredi
1093 chef_de_etat	678 premier_ministr	519 source_proche
1086 plus_grand	671 proche_orient	509 dire_m
1063 week_end	652 parti_socialiste	507 protection_de_environnement

Table 1. Samples of the terms built automatically

In comparison with the manual terminological base, the automatic base is less accurate. That is, many established items are not real terms. For example: 1086 plus_grand (larger), 724 déclarer_m (declare_me), 578 également_été (also been). These terms have been retained because of word ambiguities. For example, "plus" and "grand" may also be a noun in French, although they are respectively adverb and adjective in this example. Despite the noise, the automatic term base has a good coverage of the corpus. Thus, it is a good complement to a manual base. We used this base to recognize and extract terms from texts just in the same way as shown in Figure 1.

2.3. Retrieval results and discussion

In our experiments, we used a modified version of SMART system [2]. Prior to TREC6 experiments, we had a French corpus - OFIL - with a set of evaluated queries. This corpus is set up in the Amaryllis project [8]. It contains a set of articles published in the French journal "Le

Monde". We used this corpus to determine some variables for French IR as follows.

We first compared different stemming approaches: transforming plural to singular, and removing several sets of word endings. Finally, we chose the set of 123 word endings which gave the best results for OFIL.

We compared approaches using different combinations of terms and words: terms only, terms with words in a single vector and terms and words in two different vectors. The second approach gave the best results. Using terms only is not as good as the classical approach using words only, and separating terms from words is not as good as grouping them together. This shows that the terms alone do not have a good coverage of document contents. They have to be supplemented by single words.

We used *ltc* term weighting of SMART in all our experiments. Three sets of results have been submitted. Run2 (CLIPS2) is the result with the classical approach using (stemmed) words only. Run1 (CLIPS1) uses both the BTQ terms and words in a single vector. Run3 (CLIPS3) uses the BTQ terms, automatically built terms and words. Table 2 gives a comparison of these approaches for French to French retrieval.

Run	Run2	Run1	Run3
Total number of documents over all queries			
Retrieved:	21000	21000	21000
Relevant:	1239	1239	1239
Rel_ret:	1009	989	983
Interpolated Recall - Precision			
at 0.00	0.6859	0.6932	0.7113
at 0.10	0.5505	0.5949	0.6016
at 0.20	0.4650	0.4887	0.5028
at 0.30	0.4078	0.4072	0.4381
at 0.40	0.3644	0.3481	0.3823
at 0.50	0.3245	0.3109	0.3427
at 0.60	0.2886	0.2741	0.3061
at 0.70	0.2453	0.2331	0.2418
at 0.80	0.2173	0.1930	0.1943
at 0.90	0.1369	0.1307	0.1332
at 1.00	0.0288	0.0350	0.0401
Average precision (non-interpolated) over all rel docs			
	0.3171	0.3204	0.3404
		(1.04%)	(7.35%)
Precision:Precision:			
At 5 docs:	0.5048	0.5238	0.5714
At 10 docs:	0.4619	0.4714	0.5429
At 15 docs:	0.3841	0.4159	0.4698
At 20 docs:	0.3619	0.3857	0.4214
At 30 docs:	0.3222	0.3524	0.3540
At 100 docs:	0.2138	0.2229	0.2162
At 200 docs:	0.1493	0.1505	0.1507
At 500 docs:	0.0822	0.0783	0.0781
At 1000 docs:	0.0480	0.0471	0.0468
R-Precision (precision after R (= num_rel for a query) docs			
Exact:	0.3562	0.3439	0.3625

Table 2. Evaluation of French-French retrievals

We observe that using the terms of the BTQ together with words does not lead to a significant improvement in effectiveness. However, when the automatically built terms are also considered, an improvement of 7.35% is obtained. This suggests that, in our case, the automatically built terms have more impact on IR effectiveness than the manually established terms in the BTQ. Part of the reasons for this may be the following:

- The BTQ is not a general terminological base (as our test corpus is). Its aim is to suggest standard French terms to designate concepts in different specialized domains, as well as their correspondence with English terms. Many words in everyday language are not covered. For example, the word "mariage" (marriage) is not in the BTQ. This word designates the key concept in Query 2. So using the BTQ, we were not able to recognize a good part of important concepts in the queries and in the documents. On the other hand, some secondary concepts have been recognized as terms, for example "taux" (rate) in Query 2. When the recognized BTQ terms are combined with the words occurring in the original query (or document), the representation obtained often derive from the original content. So the retrieved documents do not correspond to the original query.

- The automatically built terms are closer to the test corpus than the manual base in the sense that all the built terms actually occur in the corpus, and frequently used terms are usually recognized. Although much noise is obtained, there is much less silence than when the manual base is used. This is why we did make some improvement in Run3.

In comparison with other experiments on French to French retrieval, our approaches performed reasonably well. Below is a comparison with the medium performance for our three runs:

Run	≥ medium	< medium
CLIPS2	13	8
CLIPS1	12	9
CLIPS3	14	7

Table 3. Comparison with the medium performance

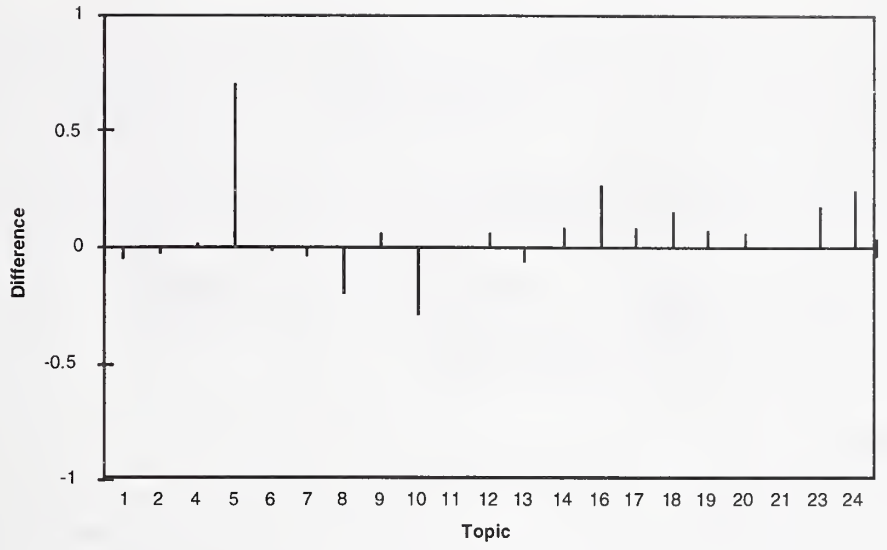
Figure 2 shows more details of the comparison for each query.

2.4. Cross-language retrieval

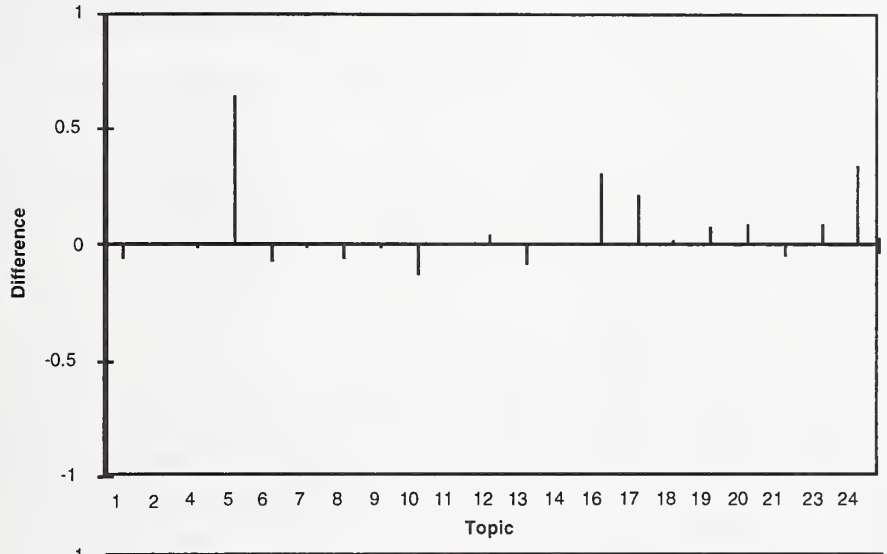
At the time of official submission of retrieval results, our implementation for cross-language retrieval was not finished. So no result was submitted. The implementation was completed later on. We tested the effectiveness of the BTQ for English to French retrieval (i.e. using English queries to retrieve French documents). As a point of comparison, we also used LOGOS™ translation system to translate the English queries into French, and then using the classical IR approach to retrieve documents. We obtained an average precision of 20.83% by this approach.

For cross-language retrieval using the BTQ, documents in French are indexed using both words and the BTQ French terms. English terms are extracted from the queries in a similar way to term extraction from documents. They are translated into their French equivalents by the BTQ, which are then used to retrieve French documents.

Run2:



Run1:



Run2

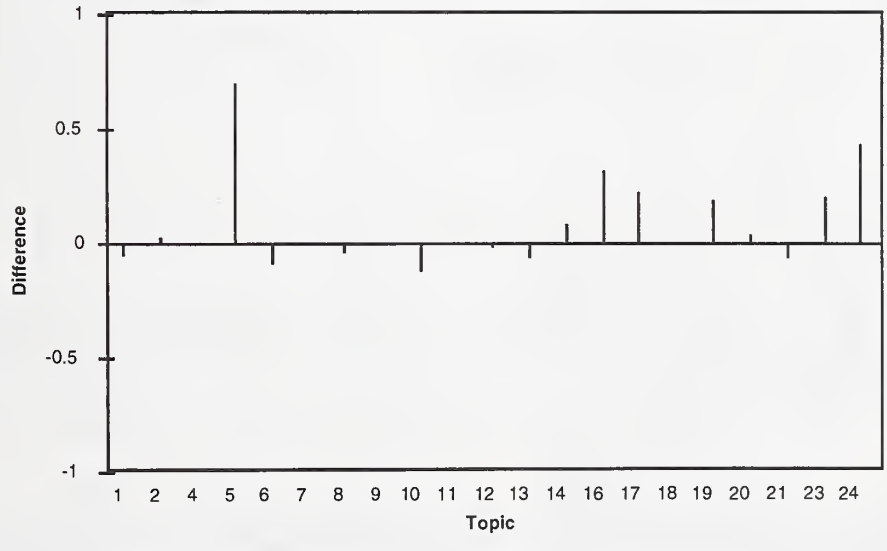


Figure 2. Comparison with the medium performance

Although this approach sounds valid in principle, we did not obtain many important concepts from the queries. We mentioned the problem with the word "mariage" in French for Query 2. This same problem also occurs for the English part of the BTQ. In addition, due to the language barrier, we could not combine the French BTQ terms with the English words in the original queries in order to diminish the silence in the representation (although the Cornell group made a surprising success in using English words to retrieve French documents, see in the same proceedings). Due to these facts, our use of the BTQ for cross-language retrieval led to a very low average precision of 7.98%. Our conclusion from this experiment is that the BTQ is not appropriate for general domain cross-language retrieval. It is still questionable whether it helps in cross-language retrieval in specialized domains.

3. Chinese IR

The difference between Chinese IR and IR for European languages lies in the fact that words are not separated in Chinese sentences. For example, the phrase "information retrieval system" is written as 信息检索系统. It is important to separate a sentence into smaller segments. Two types of segments may be used: N-grams or words.

3.1. Using N-grams

An N-gram is a subsequent string of N Chinese characters. For example, the string 信息检索系统 (information retrieval system) may be segmented into the following unigrams (N=1) and bigrams (N=2):

N=1: 信 息 检 索 系 统

N=2: 信息 息检 检索 索系 系统

In the case of bigrams, it is possible to consider the combinations of the first and the last characters with sentence boundaries or punctuation particles, that is _信 and 统_. However, punctuation particles and sentence boundaries are meaningless for IR purposes. We believe that such bigrams will have little impact on retrieval effectiveness. So we do not consider them.

It is possible to use longer N-grams for Chinese IR. However, it has been shown that bigrams are a good choice for Chinese IR [6].

3.2. Using words

This approach requires one to segment a Chinese sentence into words. This is not a trivial task because of the enormous amount of ambiguity. A sentence may often be segmented into several different sequences of legitimate words. For example, the sentence 现在本所有究生活动 (there is currently an activity for graduate students in our institute) contains the following legitimate words:

现 (now), 现在 (now),
在 (at),
本 (originally), 本所 (our institute),
所 (institute), 所有 (all, belong to),
有 (have),

研究 (research),	研究生 (graduate students),
生 (give birth),	生活 (life),
活 (live),	活动 (activity),
动 (move).	

There are as many as 30 possible combinations of legitimate words. Only the following one is correct: 现在 本所 有 研究生 活动 (now / our institute / have / graduate student / activity). The key problem is to choose the correct segmentation among all the possible solutions.

There are two basic segmentation approaches for Chinese: the approach based on a dictionary, and the approach based on statistics (see [7] for discussions).

In the dictionary-based approach, one first finds all the legitimate words included in a sentence, then the longest-matching algorithm is applied to choose the sequence of words which covers the sentence with the longest words (or with the fewest words). A dictionary-based segmentation is usually augmented by a set of heuristic rules to recognize special sequences such as quantity-classifier sequences (e.g. 一千个 - one thousand [units of]).

On the other hand, a statistical approach relies on statistical data to determine possible words and to select the best word sequence. Statistical data are usually obtained from a set of manually segmented training texts. According to the frequency of occurrences and co-occurrences, one may determine how probable a string (possibly within some context) may be a word.

From the point of view of performance, the previous experiments showed that the two approaches have comparable accuracy. In our experiments, we used the dictionary-based word segmentation because no training text from this corpus was available. In addition of a word dictionary of 87 600 entries, we also dealt with some special character sequences which may be considered as words in Chinese. These sequences include: nominal pre-determiner and affix structure. A set of rules are set up for their recognition. For example, ２０００年 (year 2000), 第一回 (first time) are recognized as nominal pre-determiners, and "小朋友" (little friend) and "大众化" (popularize) as having an internal affix structure.

One particular problem we dealt concerns numbers. Numbers may be written in different ways in Chinese texts. For example, "the year 2000" may be written in Arabic numbers which may be encoded in ASCII (2000年) or in Chinese codes (２０００年). It may also be written in Chinese numbers as 二零零零年, or as a mixture of Chinese and Arabic numbers: 二０００年. Some of the queries contain numbers (e.g. a date). It is important to normalize them so that the same number becomes identical in documents and in queries. In our segmentation, a normalization is performed.

3.3. Particularity of segmentation for IR: Long vs. short words

When a dictionary is used, the maximum-matching algorithm is usually applied. However, as there is no clear definition of words in Chinese, in many Chinese dictionary, there are a number of long words/phrases that are composed of shorter words. For example:

long words	component words
环境污染 (environment pollution):	环境 (environment), 污染 (pollution)
安全措施 (security measure):	安全 (security), 措施 (measure)
电脑网络 (computer network):	电脑 (computer), 网络 (network)

If a long word/phrase is encountered, the shorter words contained in it are hidden. If we consider the segmentation problem from a different standpoint than IR (e.g. Machine Translation), this is not problematic. However, it will lower the recall ratio in IR. For example, if a document talks about 环境污染 (environment pollution) and a query asks for "污染" (pollution), the document will not be retrieved. In order to avoid this problem, our segmentation process extracts all the possible compound words (composed of two characters or more) from a given character string. So for the sequence 环境污染, three words will be extracted: 环境污染, 环境 and 污染. In one of our official runs submitted, this approach is used: all the compound words included in a long word are also extracted.

This approach may be further extended by also extract all the single-character words. So, for the string 环境污染, we also have the following segments extracted: 信, 息, 系, 统. This approach has been tested after our official submissions.

3.4. Experiments

We submitted two runs: one using bigrams and another using words. The answers submitted are also retrieved with SMART system. Notice that once the documents and the queries have been cut down into segments, SMART system may be used for their indexing and retrieval, after some modifications. A stop-list of 1460 elements is set up. This list contains frequently used functional words as well as symbols. The functional words included are usually prepositions, adverbs, tense particles and so on.

The evaluation of the two approaches is shown in Table 4. We observe that there is no significant difference between the two runs with regard to effectiveness. For most queries, both approaches perform quite well. However, for a few queries (CH33 and CH34 in particular), the average precision is quite low (from 0.0609 to 0.1320). This is the same for the other groups, too. In particular, Query CH33 is a difficult one for automatic IR systems. It asks for documents about particular events of airplane hijacking between mainland China and Taiwan, while documents discussing airplane hijacking at a general level (e.g. measures to prevent hijacking) are not relevant. In our top ranked answers, most documents concern airplane hijacking at a general level. A few other documents describe hijacking events in other countries.

We also tested the impact of number normalization on IR. Query CH35 is the one which involves most nominal pre-determiners. It asks for documents about the event of a particular date. Without normalization, we obtained 0.4663 for average precision. With the normalization, we obtained 0.5271. This shows that the normalization is necessary when querying documents with numbers.

After the official submissions, we tested the extended word segmentation approach by also extracting single characters. Using this approach, we obtained a better average precision of 46.15%. Still, this performance is not significantly different from that using bigrams. The reason possibly lies in the Chinese language itself: in Chinese, single characters (ideographs) may constitute a reasonably good representation of a text. In order to confirm this fact, we indexed the documents and the queries only by single characters. We obtained a quite high performance of 41.09%. Similar result is also obtained in [6]. This result clearly shows that the good performances obtained for both bigram- and word-based approaches is due in major part to the meaningfulness of Chinese characters. Then a possible explanation to the almost identical performances for both bigram- and word-based approaches is as follows: Both

Approach:	Bigrams	Words
Total number of documents over all queries		
Retrieved:	26000	26000
Relevant:	2958	2958
Rel_ret:	2709	2668
Interpolated Recall - Precision Averages:		
at 0.00	0.8546	0.8581
at 0.10	0.6844	0.7134
at 0.20	0.6165	0.6582
at 0.30	0.5556	0.5953
at 0.40	0.5236	0.5345
at 0.50	0.4815	0.4939
at 0.60	0.4252	0.4371
at 0.70	0.3711	0.3609
at 0.80	0.3043	0.2608
at 0.90	0.1912	0.1541
at 1.00	0.0284	0.0169
Average precision (non-interpolated) over all rel docs		
	0.4467	0.4524 (1.28%)
Precision:		
At 5 docs:	0.6615	0.6769
At 10 docs:	0.7000	0.6423
At 15 docs:	0.6359	0.6333
At 20 docs:	0.5981	0.6173
At 30 docs:	0.5705	0.5782
At 100 docs:	0.4408	0.4662
At 200 docs:	0.3379	0.3402
At 500 docs:	0.1899	0.1857
At 1000 docs:	0.1042	0.1026
R-Precision (precision after R docs retrieved):		
Exact:	0.4655	0.4748

Table 4. Comparison of the retrieval performances

bigrams and words allow to enhance the document and query representation in comparison with single characters. However, the enhancement is more limited in both cases than we could expect (increasing the average precision by about 10%). So the comparable performances of bigrams and words are due to their limited capacity to improve the document and query representation by single characters.

In comparison with the results of other groups, ours are not quite good. Table 5 shows the comparison of our results with the medium performance.

Run	≥ medium	< medium
Bigrams	6	20
Words	6	20

Table 5. Comparison with the medium performance

The reason of these poor performances is that we did not use any further techniques in our runs to improve the effectiveness, as several other groups did. Our goal in these tests is to compare

Chinese IR using bigrams and words on the same basis. By using some techniques such as using the top ranked documents to do feedback retrieval as well as combining bigrams with words, the effectiveness may be increased.

3.5. Time and space

It is also important to compare the two approaches with respect to the time and space requirements. Table 6 shows the comparison.

	Bigrams	Words
NB. of tokens	1 446 354	205 056
Segmentation time	1 h 04 mn	4 h 56 mn
Indexing time	13 h 12 mn	1 h 19 mn
Total document processing time	14 h 16 mn	6 h 15 mn
Retrieval time	7 mn	5 mn
Space	1.2 Gb	0.5 Gb

Table 6. Time and space for IR using bigrams and words

This table shows that the total processing time for Chinese IR using bigrams are more than twice longer than that using words. So is the comparison on the space needed for their indexing. We can conclude that on the time and space criterion, IR using words has advantages over IR using bigrams.

4. Conclusions

We compared an IR approach based on terms with the classical word-based approach in French and cross-language IR. For French IR, it is shown that terms, together with words, may bring improvements to the system performance. We noticed that using a manually established terminological base leads to little improvement over the classical approach. However, using both the manual base and the automatic base brings a significant improvement. The little improvement by using the manual terminological base is due to its poor coverage of the test corpus. Many important concepts are not recognized. By incorporating an automatically built term base, the coverage is increased. So, we were able to achieve a better performance.

The manual terminological base is also used in the cross-language retrieval after the official submission The result is disappointing. It is much lower than those obtained by the other groups (which is about 50% of the monolingual IR). Again, the primary reason is the poor coverage of the terminological base used. In particular, in this case, the base is left alone to recognize the concepts from the queries (without the help of words). The poor coverage has complete impact on the global result of this test.

For Chinese IR, two approaches have been compared: one is based on bigrams and the other on words. From the point of view of effectiveness, there is no significant difference between the two approaches. However, if we also consider time and space, the approach based on words has some advantages. So, despite the comparable effectiveness, we still advocate the word-based approach if one has to chose between them. Moreover, it seems much easier to do cross-language

retrieval with words than with N-grams. We also believe that the word-based approach has not reached its limits. Improvements are still possible, for example, by integrating more heuristic rules to recognize more special sequences, or by incorporating a thesaurus. This is part of our future work.

Acknowledgment: This research is partly supported by a grant for France-Quebec research cooperation in linguistic engineering.

References

1. M.-F. Bruandet, Outline of a knowledge base model for an intelligent information retrieval system. *Information Processing and Management*, vol. 25, pp. 89-115 (1989).
2. C. Buckley, Implementation of the SMART information retrieval system. Cornell University, Technical report 85-686, (1985).
3. J. Fagin, Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic methods. in *Computer Science*: Cornell University, 1988.
4. V. Güntzer, G. Jüttner, S. G., and F. Sarre, Automatic thesaurus construction by machine learning from retrieval sessions. *Information Processing & Management*, vol. 25, pp. 265-273 (1989).
5. H. Kimoto and T. Iwaderie, Construction of a dynamic thesaurus and its use for associated information retrieval. *13th ACM-SIGIR Conference*, 227-240 (1990).
6. K. L. Kwok, Comparing representations in Chinese information retrieval. *ACM-SIGIR'97*, 34-41 (1997).
7. J.-Y. Nie and M. Brisebois, On Chinese text retrieval. *ACM-SIGIR'96*, Zürich, 225-233 (1996).
8. J.-Y. Nie, J.-P. Chevallet, and Y. Chiaramella, Vers la recherche d'information à base de termes. *1er Journéees Scientifiques et Techniques FRANCIL*, Avignon, 119-125 (1997).
9. C. J. v. Rijsbergen, A theoretical basis for the use of co-ocurrence data in information retrieval. *Journal of Documentation*, vol. 33, pp. 106-119 (1977).
10. K. Sparck-Jones, Notes and references on early automatic classification work. *SIGIR Forum*, vol. 25, pp. 10-17 (1991).

Interactive Retrieval using IRIS: TREC-6 Experiments

Robert G. Sumner, Jr., Kiduk Yang, Roger Akers, and W. M. Shaw, Jr.
School of Information and Library Science
University of North Carolina
Chapel Hill, NC 27599-3360 USA
{sumnr, yangk, shaw}@ils.unc.edu
akers@unc.edu

0 Submitted Runs

unc6ia and *unc6ip* – interactive track runs
unc6ma – Category B, manual adhoc task run
unc6aal – Category B, automatic adhoc task run (long query)
unc6aas – Category B, automatic adhoc task run (short query)

1 Introduction

For the TREC-5, Category B adhoc task, we examined the effectiveness of two relevance feedback models: an adaptive linear model and a probabilistic model (Sumner & Shaw, 1997). The models were shown to be effective, especially when the relevance assessments of the searchers matched those of the official TREC judges. During feedback, the query was expanded by a large number of terms from the retrieved documents. Some queries were expanded by as many as 1000 terms.

Building on the basic framework of our TREC-5 system, we developed an interactive, Web-based retrieval system called IRIS (Information Retrieval Interactive System[1]) for TREC-6. Although IRIS inherits both the adaptive linear and the probabilistic model from the TREC-5 system, we made significant modifications to the implementation of both models in order to use a three-valued scale of relevance during feedback. Furthermore, we expanded the scope of human interaction with the system. For example, throughout the search process, the searcher can add and delete query terms as well as change their weights. Moreover, statistically significant, two-word collocations have been added to the term index. IRIS uses collocations not only in formulating the feedback query, but also in presenting to the searcher "suggested phrases" (i.e., collocations related to the initial query), prior to the first document retrieval pass. Finally, as with our TREC-5 system, during feedback the query is expanded by a large number of terms. However, for reasons of efficiency, the number of terms in the query was limited to 300 in our TREC-6 system.

The primary focus of our TREC-6 experiments was on the interactive track and the manual, Category B adhoc task. People were hired to conduct searches for these runs. Here, we are interested not only in the official TREC results but also (perhaps more so) in the reactions of the searchers to the various features of IRIS. The searchers' responses to questionnaires as well as the retrieval effectiveness of the searches are analyzed in this paper as we address, among other things:

- What are the relative effectiveness and the different properties of the adaptive linear and the probabilistic models? Which model do the searchers prefer?

[1] A prior version of IRIS was developed by Kiduk Yang, Kristin Chaffin, Sean Semone, and Lisa Wilcox at the School of Information and Library Science (SILS) at the University of North Carolina. They worked under the supervision of William Shaw and Robert Losee.

- What are the frequencies of documents declared relevant, marginally relevant, and nonrelevant by the searchers? Do searchers utilize all three categories of relevance?
- What is the effectiveness of the suggested collocations? Do searchers find them helpful?

2 Features of IRIS

The features described here apply to the interactive track and manual adhoc runs, and not necessarily to the automatic adhoc runs.

2.1 Stemming and Indexing

The full-text of 210,158 Financial Times (FT) documents was processed to generate a single-word index consisting of 401,423 terms and a collocation index of 400,576 terms. Processing of the full-text involved removing punctuation, numbers, and the 390 high-frequency terms listed in the WAIS default stopwords list. We then conflated morphological variations of words by applying "the modified Krovetz inflectional stemmer."[2]

This stemmer implements a modified version of Krovetz's inflectional stemmer algorithm (Krovetz, 1993). Our stemmer restores the root form of plural ("-s," "-es," "-ies"), past tense ("-ed"), and present participle ("-ing") words, provided this root form is in our online dictionary. The modified Krovetz inflectional stemmer was chosen over other suffix removal stemmers such as Porter's stemmer and SMART's modified-Lovins stemmer, in part due to its conservative approach to stemming. In our TREC-5 experiments (Sumner & Shaw, 1997), we felt that SMART's stemmer incorrectly stemmed too many words and thus had a detrimental effect on precision. For example, "Spence," "Spencer," and "spent" all stemmed to "spent," and "Alger" and "algae" both stemmed to "alg."

2.2 Collocation Index

To augment single-word terms, two-word collocations were automatically extracted from the collection and used to generate a second index. A collocation is defined loosely as a pair of terms that occur together more frequently than normally expected. For descriptive purposes, a collocation consists of a "target word" and its "collocate." Do these frequently co-occurring terms represent a concept or "meaning" which can be automatically extracted and used in information retrieval? "Collocational meaning" is discussed in linguistics and has been investigated for utility in lexicographical tasks (Choueka, Klein, & Neuwitz, 1983; Firth, 1957; Smadja & McKeown, 1990). Our system attempts to take advantage of this collocational meaning to provide a finer level of discrimination between documents.

The collocation indexing process began by extracting from the stemmed collection (without stopwords) all two-word pairs occurring within ±3 words of each other in a paragraph (Haas & Losee, 1994; Losee, 1994; Martin, Al, & van Sterkenburg, 1983; Phillips, 1985). This process generated a very large list of possibly meaningful collocations. It is obvious that using a word window of ±3 words over a collection of documents will result in the extraction of pairs of words that co-occur purely by chance and have no useful syntactic or semantic relationship to each other. Therefore, the next step in creating a useful supplemental index was to cull the list leaving only those two-word pairs co-occurring with outstanding frequency. A z-score representing the probability of the two words co-occurring by chance was calculated for each pair in the list (Berry-Rogghe, 1974). This probability was based on the frequency distribution of the individual terms and the term pairs. All word pairs with a z-score of 2.576 or greater ($\alpha = 0.005$) were considered to co-occur with statistically significant frequency. The final collocation index consisted of statistically significant collocations that occurred more than one time in the collection.

2.3 Ranking Function and Document Term Weights

Documents are ranked in decreasing order of the inner product of document and query vectors,

[2] This stemmer was developed by Kiduk Yang, Danqi Song, Woo-Seob Jeong, and Rong Tang at SILS at UNC.

$$\mathbf{q}^{\mathrm{T}}\mathbf{d}_i = \sum_{k=1}^{t} q_k d_{ik}, \qquad (1)$$

where q_k is the weight of term k in the query, d_{ik} is the weight of term k in document i, and t is the number of terms in the index. Document term weights are SMART *Lnu* weights, which were effective in both TREC-4 (Buckley, Singhal, Mitra, & Salton, 1996) and TREC-5 (Buckley, Singhal, & Mitra, 1997). According to Singhal, Buckley, and Mitra (1996), *Lnu* weights were created in an attempt to match the probability of retrieval given a document length with the probability of relevance given that length. Our implementation of *Lnu* weights was the same as that of Buckley et al. (1996, 1997) except for the value of the "slope" in the formula, which is an adjustable parameter. The optimal value for slope may depend, in part, on the properties of the document collection. Based on test runs using TREC-5 topics, we used a slope of 0.3 for the FT collection for both the initial search iteration and feedback iterations. Unfortunately, as explained later, there was a bug in these test runs.

2.4 Initial Query Formulation

Figures 1 through 5 all show different screens in IRIS for the same search. In the search the user is interested in the various drugs used to treat asthma. (This topic was used as an example in NIST's interactive track tutorial for the control system, ZPRISE.)

Figure 1 shows the screen in IRIS where the user enters the initial query for the search. The user can "emphasize" a term in the query by adding an asterisk to the end of it. The user can also enter two-word collocations (or "phrases") in two different ways. To indicate that the query should include, not only the collocation, but its component words as well, the user should enclose the collocation in double quotes. To indicate that the query should not include the component words, the user should enclose the collocation in single quotes.

After the user clicks the "Search" button, IRIS removes stopwords from the query, stems words, and computes SMART *ltc* query term weights (Buckley, C., Salton, G., Allan, J., & Singhal, A., 1995). It also adds 1.0, the maximum possible *ltc* weight, to the *ltc* weights of terms which were emphasized by the user. An "Initial Query Modification" screen is then displayed (see Figure 2). The stemmed term, the number of postings, and the query term weight (multiplied by 10 and rounded off for ease of reading) are listed for each term entered by the user. Terms that are not in the collection's index are not displayed. Since *ltc* weights incorporate inverse document frequency, the weights are inversely proportional to the number of postings. Also, note that "asthma" has a high weight because it was emphasized (see Figure 1). The user can change these query term weights if she wishes. The user can also further modify the query by going back to the previous screen using her Web browser. Alternatively, she can formulate a completely new query by hitting "New Search" at the bottom of the screen.

2.5 Suggested Collocations

At the right of the screen in Figure 2 are collocations "suggested" by IRIS. If the user wishes to add any suggested collocations to her query, she can do so by changing its weight from the preset value of zero. The process by which these "suggested phrases" are chosen by IRIS is now described. The original query posed by the searcher goes through a pre-retrieval process. All two-word collocations found in the original query are extracted following the same extraction procedure used to generate the collocation index. Those collocations from the query considered "significant" in the collection (i.e., those in the collocation index) are placed at the top of the suggested phrase list. In addition, each single query term is used to look up other significant collocations that contain the query term in question. Those collocations are added to the suggested phrase list using the following rules. Collocations that include an emphasized query term are added first to the list. The collocations are then ordered according to the number of query terms with which a collocate pairs. For example, given the initial query in Figure 1, if "tilade" forms a significant collocation with both "asthma" and "drug," then "asthma tilade" and "drug tilade" will be ordered higher on the list than "drug abuse" because "abuse" only collocates significantly with "drug." Finally, the collocations are ordered by decreasing z-score. The top 30 collocations in the list are then presented to the searcher.

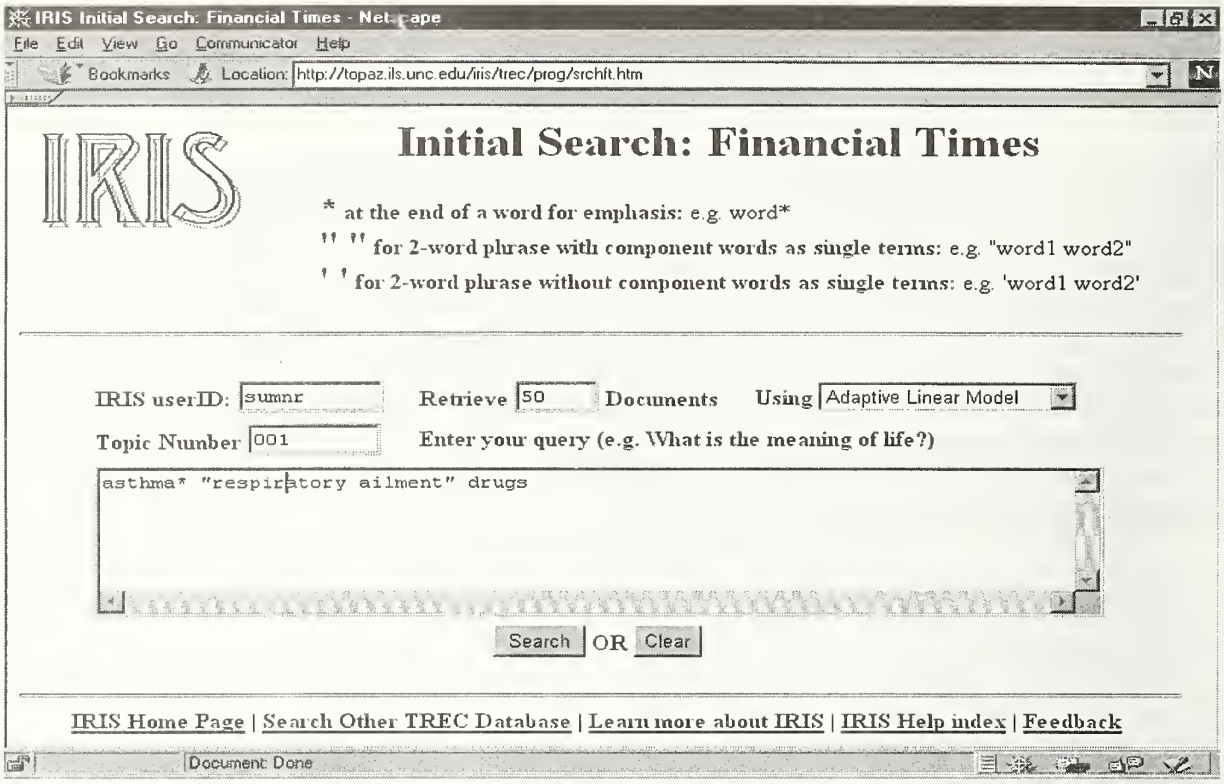


Figure 1: "Initial Search" screen of IRIS.

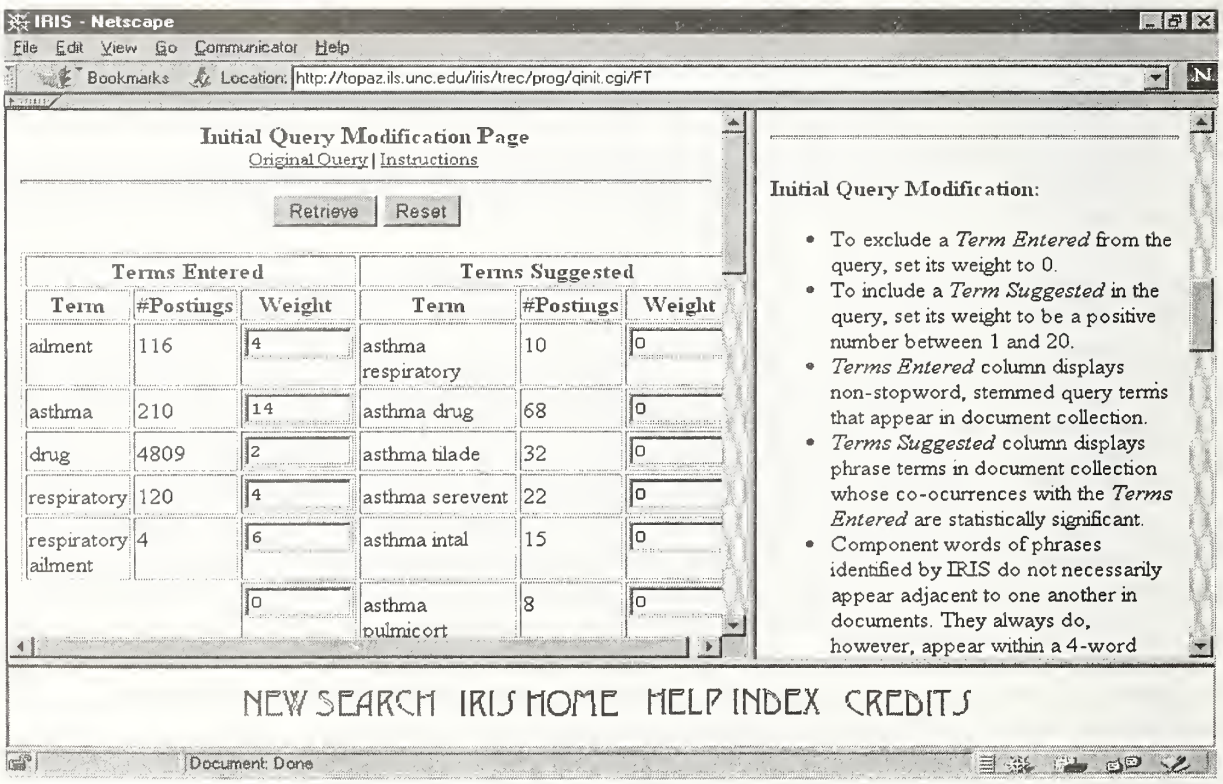


Figure 2: "Initial Query Modification" screen.

2.6 Relevance Feedback

2.6.1 IRIS Features

Figure 3 shows how retrieved documents are displayed in IRIS. Here, the initial ranking of documents is shown. The titles of the retrieved documents are displayed, and they are ranked in decreasing order of the inner product of document and query vectors. If the user clicks on the system-assigned document number to the left of a title, the corresponding document will appear in the frame on the right. The original query terms are boldfaced and the 10 highest-weighted terms in the feedback query vector are italicized in the displayed documents.

The user can utilize relevance feedback to try to improve the search. The user has the option of assigning one of three levels of relevance to a document. In the online instructions for IRIS, it is noted that the "Maybe" category could also be interpreted as "marginally relevant." An additional option is "SAVE," which was added to IRIS specifically for the interactive track. If the user selects "SAVE," the document is designated as "relevant," and it is also added to a system log indicating that the document addresses a new aspect of the query. Finally, the user may forgo the feedback process, if she wishes, by hitting "New Search" or by going back to a previous screen using the Web browser.

There may be cases where only part of a document is relevant to the query, or where a document passage contains words that the user feels should be given high weights in the next iteration of the search. In these cases, the user may wish to use the "Emphasize Terms Box" (see Figure 4). A new browser window is opened and the user can copy and paste into this window a document passage. As with the initial query, the user can indicate the special importance of a term by using an asterisk. She can also signify a collocation using either double or single quotes. Terms added to the query using the Emphasize Terms Box are stemmed and their weights are incremented by the maximum term weight of the feedback query vector. Weights of terms modified by an asterisk are incremented by twice the maximum term weight of the vector.

After making relevance assessments, the user can enter "Resubmit" as shown in Figure 3, and the designated feedback model will produce a query vector consisting of both single-word terms and collocations (see Figure 5). By default, the 25 terms with the highest positive weights and the 25 terms with the lowest negative weights are displayed.

The user can change these weights. Also, the user can add terms to the query. (In the figure, the term "boots" is added because of a pharmaceutical company by that name.) Finally, the user enters "Retrieve" to re-rank the documents.

The query vector produced by the feedback model may contain more than a thousand terms. However, query vectors of this size substantially increase the time it takes to retrieve the documents. Hence, the query vector used in the ranking process was restricted to the 250 terms with the highest positive weights and the 50 terms with the lowest negative weights.

2.6.2 Adaptive Linear Model

One of the relevance feedback models used in our experiments is the adaptive linear model (Bollmann & Wong, 1987; Wong & Yao, 1990; Wong, Yao, & Bollmann, 1988; Wong, Yao, Salton, & Buckley, 1991). This model is based on the preference relation, a concept from decision theory (Fishburn, 1970). Let $\mathbf{D}$ be the set of document vectors for a collection of documents. Then the *user preference relation* $\succ$ on $\mathbf{D}$ is defined as a binary relation on $\mathbf{D}$ where for all $\mathbf{d}_i, \mathbf{d}_j \in \mathbf{D}$,

$$\mathbf{d}_i \succ \mathbf{d}_j \Leftrightarrow \text{the user with a query prefers } \mathbf{d}_i \text{ to } \mathbf{d}_j. \tag{2}$$

An IR model based on the user preference relation allows the use of a multivalued relevance scale such as the three-valued scale used in our TREC-6 experiments. In our TREC-5 experiments (Sumner & Shaw, 1997), the adaptive linear model was used with a binary scale of relevance.

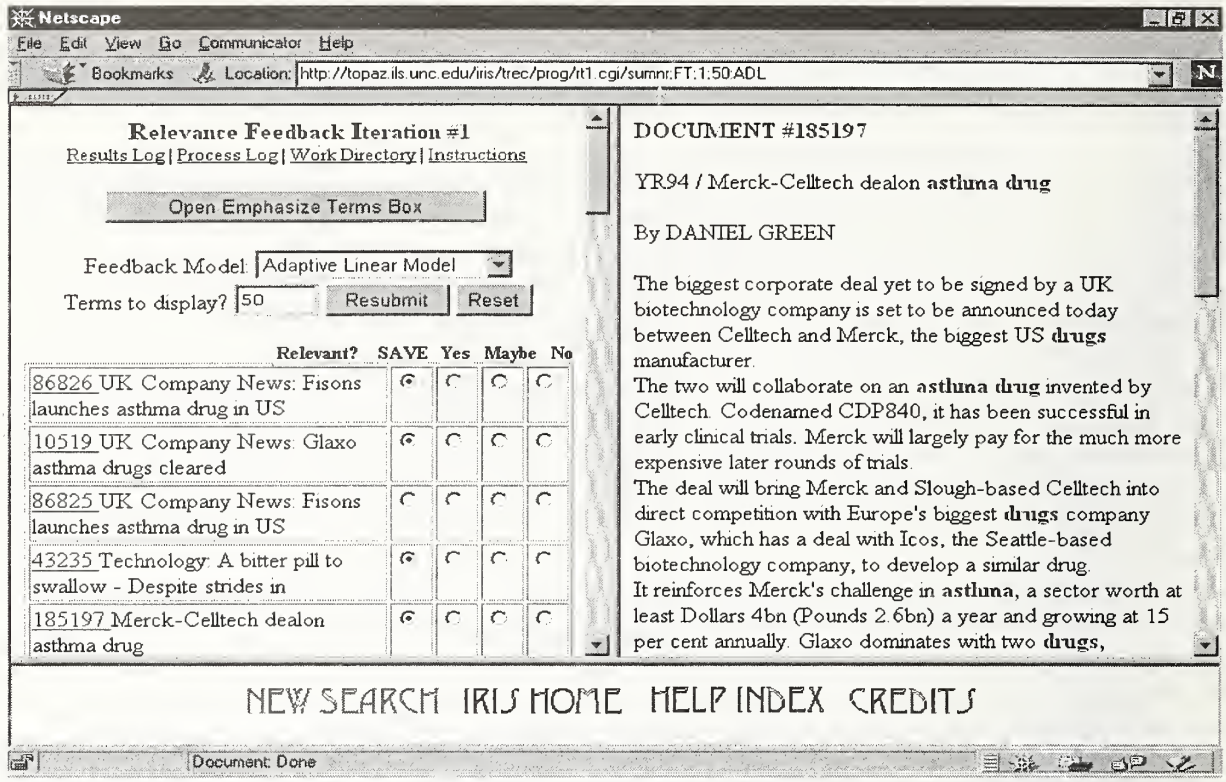


Figure 3: Ranking of documents in IRIS with the text of one of the retrieved documents displayed.

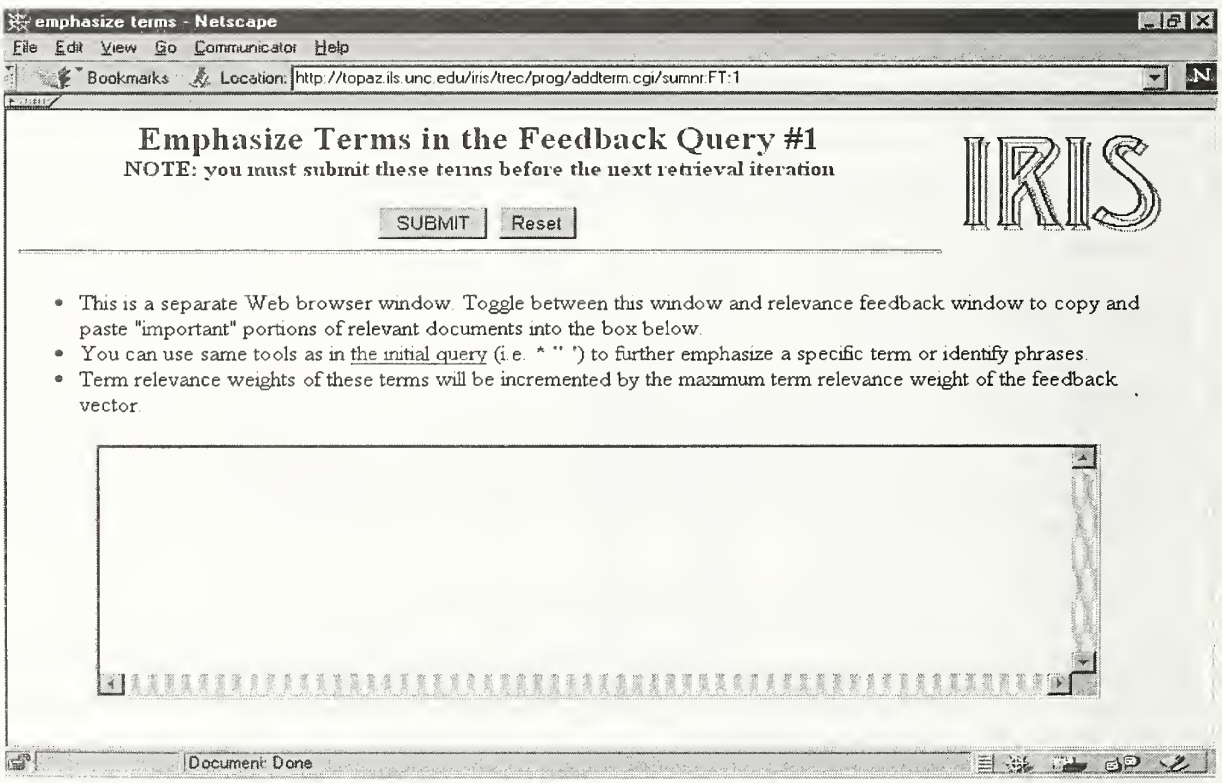


Figure 4: Emphasize Terms Box.

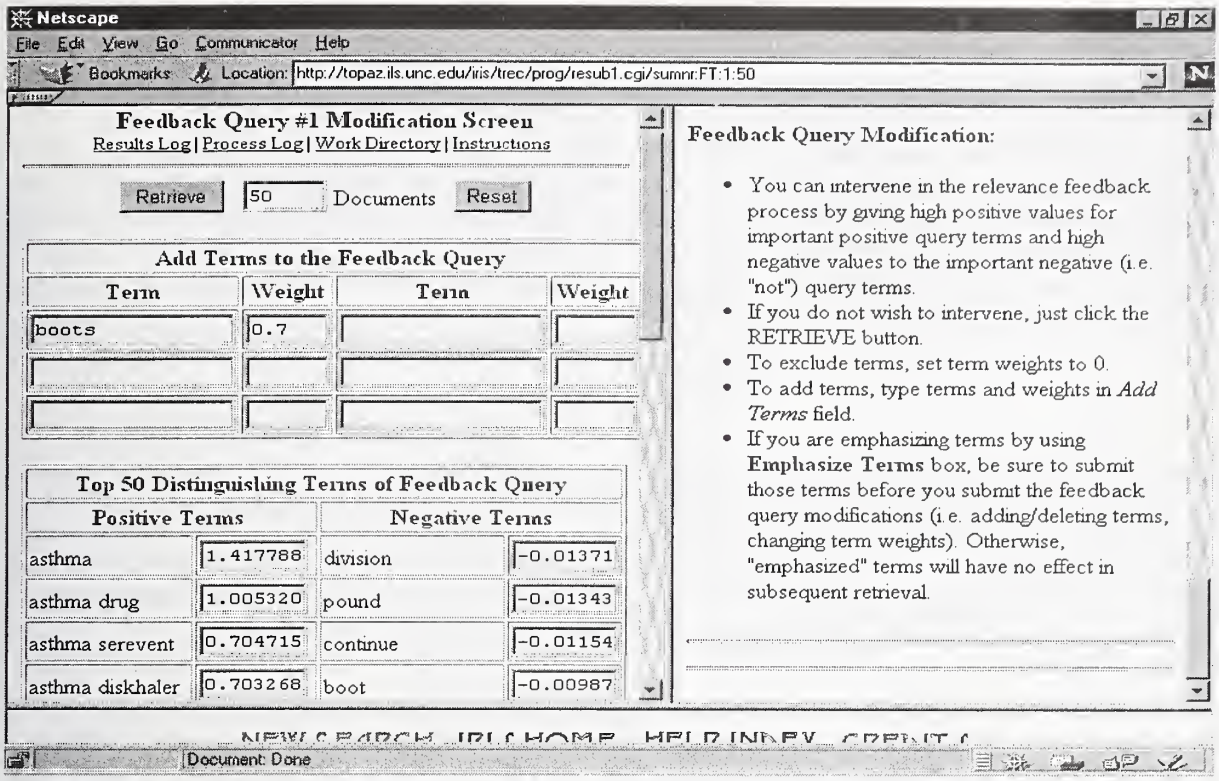


Figure 5: Feedback Query Modification Screen.

The adaptive linear model assumes that the documents in a collection are ranked according to the inner product of document and query vectors. If there exists a query vector $\mathbf{q}$ such that for all $\mathbf{d}_i, \mathbf{d}_j \in \mathbf{D}$,

$$\mathbf{d}_i \succ \mathbf{d}_j \Rightarrow \mathbf{q}^T\mathbf{d}_i > \mathbf{q}^T\mathbf{d}_j, \tag{3}$$

then this vector will always rank a more-preferred document before a less-preferred one (Wong et al., 1988). Such a query vector $\mathbf{q}$ is called a *solution vector*, and the set of all solution vectors for $\succ$ on $\mathbf{D}$ are said to comprise a *solution region* in the vector space. A solution vector can be found, provided one exists, by employing an *error-correction procedure* (Nilsson, 1965, Ch. 4; Wong et al., 1988).

However, the user's preferences are not known for the entire collection of documents. They are only known for the *training set*, which consists of those documents that have been retrieved and evaluated by the user up to that point in the search. Accordingly, in the adaptive linear model, a solution vector is found, provided one exists, for $\mathbf{T}$, the set of vectors corresponding to the training set. As $\mathbf{T}$ grows larger, one can expect that the solution region for $\mathbf{T}$ will approach that for $\mathbf{D}$ (Wong & Yao, 1990).

For our TREC-6 experiments, a solution vector for $\mathbf{T}$ was found using a variation of the error-correction procedure used by Wong et al. (1991). During a given cycle i of the algorithm, if query vector $\mathbf{q}_{(i)}$ is a solution vector, the algorithm terminates. If $\mathbf{q}_{(i)}$ is not a solution vector, a new query vector $\mathbf{q}_{(i+1)}$ is created by

$$\mathbf{q}_{(i+1)} = \mathbf{q}_{(i)} + \alpha\mathbf{b}_{max} \tag{4}$$

where α is a positive constant, $\mathbf{B} = \{\mathbf{b} = \mathbf{d}_i - \mathbf{d}_j \mid \mathbf{d}_i, \mathbf{d}_j \in \mathbf{T} \text{ and } \mathbf{d}_i \succ \mathbf{d}_j\}$, and $\mathbf{b}_{max} \in \mathbf{B}$ such that for all $\mathbf{b} \in \mathbf{B}$,

$$-\mathbf{q}_{(i)}^T\mathbf{b}_{max} \geq -\mathbf{q}_{(i)}^T\mathbf{b}. \tag{5}$$

It can be shown that this algorithm will converge to a solution vector if one exists (Nilsson, 1965, pp. 85-87).

Thus, during a given cycle of this algorithm, one document vector is added to the query vector, and another document vector, less relevant than the first one, is subtracted from it. Also, because the desired result is $\mathbf{q}_{(i)}{}^{T}\,\mathbf{b} > 0$ for all $\mathbf{b} \in \mathbf{B}$, then the quantity $-\mathbf{q}_{(i)}{}^{T}\,\mathbf{b}$ can be viewed as a measure of the extent to which $\mathbf{b}$ is in error. The vector $\mathbf{b}_{max}$ then is that $\mathbf{b}$ that produces the maximum error (Wong et al., 1991).

The *starting vector* $\mathbf{q}_{(0)}$ is the initial query vector of the error-correction procedure. The choices made for the starting vector and for the constant α are important because they influence the composition of the solution vector produced by the procedure. These choices may also influence the number of cycles that the procedure runs through before finding a solution vector.

Initially in our research, following Sumner and Shaw (1997), α was 1 and the starting vector was

$$\mathbf{q}_{(0)} = \mathbf{q}_{rk} + \sum_{new\,rel} \mathbf{d}, \tag{6}$$

where $\mathbf{q}_{rk}$ is the query vector that produced the current ranking of documents and where the summation is over all of the *new* relevant documents retrieved. A "new" relevant, retrieved document during a given search iteration is one that was not retrieved and evaluated during a previous iteration. Alternatively, it may also be a document that was declared either "nonrelevant" or "marginally relevant" in a previous iteration, but whose relevance was changed to "relevant" in the current iteration. Sumner and Shaw's choices for α and the starting vector were generalizations in the context of multiple feedback iterations of the choices made by Wong et al. (1991) in the context of one feedback iteration.

We conducted some searches where the interactive functionality of IRIS was tested. Using Equation 6 for the starting vector in these searches, we noticed that documents previously declared as "nonrelevant" were often still near the top of the ranking. We also noticed that documents previously declared as "marginally relevant" were at times "pushed down" a hundred documents or so. Hence, we decided to change the starting vector to *insure* that (1) the vectors of all new nonrelevant documents were subtracted from $\mathbf{q}_{rk}$ and that (2) the vectors of all new marginally relevant documents were added to $\mathbf{q}_{rk}$. (New marginally relevant and new nonrelevant documents are analogous to new relevant documents.) Hence, the following formula was used for the starting vector:

$$\mathbf{q}_{(0)} = c_0\mathbf{q}_{rk} + \frac{c_1}{N_{new\,rel}} \sum_{new\,rel} \mathbf{d} + \frac{c_2}{N_{new\,mrel}} \sum_{new\,mrel} \mathbf{d} - \frac{c_3}{N_{new\,nonrel}} \sum_{new\,nonrel} \mathbf{d}, \tag{7}$$

where c_0, c_1, c_2, and c_3 are constants; $N_{new\,rel}$, $N_{new\,mrel}$, and $N_{new\,nonrel}$ are the number of new relevant, new marginally relevant, and new nonrelevant documents respectively in the current iteration; and the summations, as in Equation 6, are over the appropriate new documents. This formula is similar to the relevance feedback formulas used by Rocchio (1971) and Salton and Buckley (1990). Our formula is adapted for a three-valued relevance scale, though, instead of a binary scale. Of course, these formulas can be generalized to any multi-valued relevance scale.

We used values of $c_0 = 1.0$, $c_1 = 1.2$, $c_2 = 0.6$, and $c_3 = 0.6$ in Equation 7. In addition, we used a value of $\alpha = 0.5$ in Equation 4. Because every vector of a new document is either added to or subtracted from $\mathbf{q}_{rk}$ in Equation 7, we thought that the value for α should be less than one to reduce the influence of any one new document. The value for c_2 is 0.6 so that a new marginally relevant document that is subtracted only one time in the error-correction procedure will contribute some to the final query vector. (In such a situation with an α of 0.5, a marginally relevant document $\mathbf{d}_j$ would contribute $0.1\mathbf{d}_j$ to the final query vector.) Finally, $c_1 = 1.2$ so that the influence of relevant documents would be double that of marginally relevant ones and $c_3 = 0.6$ for internal consistency.

Although there was still the problem of previously declared, nonrelevant documents "floating" to the top of the document ranking, it seemed to be less of a problem using Equation 7 than using Equation 6. Also, marginally relevant documents did not appear to be pushed down as frequently using Equation 7. However, we did not do a systematic investigation of these properties.

Of course, it is possible that there is not a solution vector for $\succ$ on $\mathbf{T}$. However, a solution vector usually exists for a set of document vectors like $\mathbf{T}$, where the number of vectors in the set are much less than the number of terms in the indexing vocabulary (Nilsson, 1965, pp. 32-35). Wong et al. (1991) and Sumner & Shaw (1997) found solution vectors for every $\mathbf{T}$ in their experiments—as did Sumner in an unpublished study. Problems can still arise, however, especially in the case where duplicate documents or "near-duplicates" are assigned different levels of relevance.

Searches conducted on the FT collection revealed the presence of either duplicates or near-duplicates. Accordingly, to take into account situations where a solution vector may not exist, the number of cycles in the error-correction procedure was limited to 201, and then $\mathbf{q}_{(201)}$ was returned as the feedback vector. This threshold was also chosen so that the user would not have to wait an inordinately long time for IRIS to produce the feedback vector.

Finally, even though the feedback vector produced by the adaptive linear model may be a solution vector for T, the vector actually used to rank the documents in IRIS during the next iteration of the search may not be one. Firstly, the user was allowed to change the weights of terms and also to add terms to the query. Secondly, to increase the speed of the retrieval process, the number of terms in the query vector used to rank the documents was limited to 300. This may mean that a large number of terms are excluded from the query vector. Due to query expansion during the error-correction procedure as well as during the creation of the starting vector, the feedback vector produced by the adaptive linear model may have as many as 1000, or even 5000, terms.

2.6.3 Probabilistic Model

In addition to the adaptive linear model, a variation of the binary probabilistic feedback model used in our TREC-5 experiment (Sumner & Shaw, 1997) was implemented in IRIS. Terms in the feedback query vector came from relevant or marginally relevant documents of the training set. To increase the speed of the retrieval process, the vector was limited to the 250 terms with the highest positive weights and the 50 terms with the lowest negative weights. The traditional binary relevance weight formula (Robertson & Sparck Jones, 1976), however, was modified to accommodate three levels of relevance judgments. Also, *Lnu* document term weights were used by Equation 1 to rank the documents.

The tri-level term relevance weight of term k is denoted by $(tr)_k$ and is defined by

$$(tr)_k = \log\left[\frac{p_k/(1-p_k)}{u_k/(1-u_k)}\right] + \frac{1}{2}\times\log\frac{m_k}{(1-rm_k)}, \tag{8}$$

where p_k is the probability term k appears in a relevant document of the training set, u_k is the probability term k appears in a nonrelevant document of the training set, m_k is the probability term k appears in a marginally relevant document of the training set, and rm_k is the probability term k appears in a relevant or marginally relevant document of the training set. When a term appears in all or none of the relevant, marginally relevant, or nonrelevant documents in the training set, estimations of p_k, u_k, m_k, and rm_k can lead to undefined values of $(tr)_k$ and therefore computing equations must be adjusted to estimate the probabilities in such instances. In TREC-5, we used Shaw's "alternative" computing equation (1995) to determine p_k and u_k instead of the "conventional" 0.5 formula (Robertson & Sparck Jones, 1976), which can overestimate term relevance weights when few relevant documents are detected in the training set (Shaw, 1995; van Rijsbergen, Harper, & Porter, 1981; Yu, Buckley, Lam, & Salton, 1983). Estimation of m_k and rm_k are done in a similar manner:

$$p_k = \frac{r_k}{N_r} \qquad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } r_k = 0 \\ 1-\frac{1}{N_d^2} & \text{if } r_k = N_r \end{bmatrix}, \tag{9}$$

$$u_k = \frac{d_k - r_k}{N_d - N_r} \quad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } d_k - r_k = 0 \\ 1 - \frac{1}{N_d^2} & \text{if } d_k - r_k = N_d - N_r \end{bmatrix}, \tag{10}$$

$$m_k = \frac{mr_k}{N_{mr}} \quad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } mr_k = 0 \\ 1 - \frac{1}{N_d^2} & \text{if } mr_k = N_{mr} \end{bmatrix}, \tag{11}$$

$$rm_k = \frac{r_k + mr_k}{N_r + N_{mr}} \quad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } r_k + mr_k = 0 \\ 1 - \frac{1}{N_d^2} & \text{if } r_k + mr_k = N_r + N_{mr} \end{bmatrix}, \tag{12}$$

where N_d, N_r and N_{mr} are the total number of documents, the total number of relevant documents, and the total number of marginally relevant documents, respectively, in the training set, and d_k, r_k and mr_k are the number of documents, the number of relevant documents, and the number of marginally relevant documents, respectively, in which term k appears. The alternative computing equations with binary relevance judgments have been shown to be highly effective in retrospective and predictive tests in a small retrieval test collection (Shaw, 1995, 1996), and were therefore adapted to three-valued relevance judgments for comparison purposes.

In our TREC-6 experiments, however, we inadvertently discarded the second term of Equation 8. Hence, we used the conventional binary term relevance weight formula.

The tri-level term relevance weight formula (Equation 8), as is the case with the binary term relevance weight formula, is a special case of a more general multi-level relevance formula, which is essentially a document ranking function with graded relevance judgments (Yang & Yang, 1997). It is easy to see that the tri-level term relevance weight formula collapses into the binary term relevance weight when the notion of marginal relevance is taken out. The document ranking function with graded relevance judgments can be shown to preserve the relevance rank order of documents (Yang & Yang); however, the computing formula that estimates the probabilities from the training set remains to be proven. Furthermore, the basic approach of the probabilistic model—i. e., using the training set to estimate the probabilities—risks poor performance when the training set is small, which is often the case in an operational setting.

2.7 Pre-Testing

A number of system decisions with respect to the the manual adhoc task and the interactive track were based, either entirely or in part, on pre-testing using the FT collection and TREC-5 topics. Relevance feedback was simulated automatically using official TREC relevance judgments. Retrieval effectiveness was evaluated using average non-interpolated precision and optimal F values (Shaw, Burgin, & Howell, 1997a, 1997b; van Rijsbergen, 1979).

Several decisions were based on this data. Unfortunately, a bug in these pre-testing runs made their results invalid. First, the adaptive linear model was chosen as the relevance feedback model for the manual adhoc task

over the probabilistic model and a fusion model (Lee, 1995, 1996a, 1996b). Second, a slope of 0.3 was utilized for the *Lnu* document term weights during both the initial search iteration and feedback iterations. Third, collocations were added to the feedback vectors along with single-word terms. Fourth, the query vector was limited to the top 250 positive-weighted terms and the lowest 50 negative-weighted terms. In these test runs with the bug, the best retrieval effectiveness came from the run where the number of terms in the query vector was not limited; however, we decided to limit the number of terms in order to decrease retrieval time. Finally, with respect to the adaptive linear model, the values of $c_0 = 1.0$, $c_1 = 1.2$, $c_2 = 0.6$, and $c_3 = 0.6$ were used for the starting vector in Equation 7.

Even without the bug, it would be difficult to generalize these automatically-generated results to interactive searches on IRIS. First, the official TREC relevance judgments are binary instead of three-valued. Second, in IRIS users can add terms to the feedback vector, delete terms, and change their weights. Third, it is difficult to simulate other aspects of the retrieval behavior of users such as the number of documents that are examined during a given search iteration. The great variation in searching behavior among users makes this task especially daunting.

3 Interactive Track Runs

3.1 Methodology

We submitted two interactive track runs: *unc6ia* and *unc6ip*. The adaptive linear model was employed in *unc6ia* and the probabilistic model in *unc6ip*. The four searchers for *unc6ia* were designated as *irisa1i* through *irisa4i*, and the four searchers for *unc6ip* were designated as *irisp5i* through *irisp8i*. See the Appendix for information about the searchers.

Each searcher conducted her interactive track searches during one 3 ½ to 5 hour session. She first filled out a "Pre-Study Questionnaire," from which information was gathered about her background and searching experience. She then read the "introductory instructions" from the Interactive Track Specification (Over, 1997a). She next proceeded to search on one system (either IRIS or ZPRISE) and then the other. For each system, the same sequence of events occurred. First, one of us trained the searcher on the system. An attempt was made to standardize the training, but there may have been some (mostly minor) differences between one training session and another. Second, the searcher conducted a practice search as if it were a real interactive track search. Each person searched on the same practice topic, depending on whether the system in question was the first system for that person or the second. Third, the searcher was given feedback on her practice search. Fourth, she conducted the official interactive track searches. We had suggested that she write down on a "Searcher Worksheet" at the beginning of the search, aspects that she thought may exist about a topic, and that it may be worthwhile to use the words describing these aspects in her initial query. Once she found and saved a document that covered this aspect she was to put a checkmark next to the aspect on the worksheet. Likewise, if she came across a document covering an aspect that she had not previously thought of, she was to write some words describing it on the worksheet and then put a checkmark next to those words. After each search, she filled out a "Post-Search Questionnaire," and after all three searches on the system, she filled out a "Post-System Questionnaire." Finally, after searching on both systems, she filled out an "Exit Questionnaire."

In departure from the Interactive Track Specification (Over, 1997a), we either told or implied to the searchers that they should spend the full 20 minutes on each topic. First, we did this because of our experience with the manual adhoc searches, which usually took at least thirty minutes and sometimes as long as an hour. These searches took a long time because of the searchers' thoroughness and because IRIS and the Web client-server architecture can be slow at times. It often took one or more minutes for documents to be retrieved and displayed. Second, we did it because we thought it would be easier to implement than allowing the time per search to vary. However, in hindsight, we probably should have allowed the searcher to spend no more time on a search than she wanted to. Then our searches would be more comparable to those of the other interactive track participants.

During our training session, we also gave some "hints" to searchers about how best to conduct the searches. For example, for IRIS searches, we suggested that after they save a document that covers a specific aspect of the topic, they give a high negative weight to the term in the feedback vector that best describes that aspect. Then the feedback vector will probably not retrieve documents that cover that specific aspect but may retrieve documents that cover other aspects. However, as it turned out, few searchers implemented this suggestion.

3.2 Results

Table 1 has our TREC-6 interactive track results with respect to aspectual recall, and Table 2 has the results with respect to aspectual precision. In the tables, E refers to the Experimental system (IRIS), and C refers to the Control system (ZPRISE). See Over (1997a) for a detailed explanation of the interactive track evaluation measures including how the "control-adjusted response" (E–C) is estimated.

Tables 1 and 2 include the mean of the six estimates of E–C for aspectual recall (aspectual precision), the standard deviation of the six estimates, the 95% confidence interval for mean E-C, the mean of the aspectual recall (aspectual precision) values for the twelve searches using E, and the mean of the aspectual recall (aspectual precision) values for the twelve searches using C. The two values given for standard deviation and the confidence interval correspond to two different ways that E-C can be estimated. For some of the measures, the run's rank with respect to the nine other interactive track runs is given.

Table 1: Aspectual recall results. Rank among 10 interactive track runs is given in parentheses.

Run	Mean of Six E–C Estimates	Standard Deviation E–C*	95% Confidence Interval for Mean E–C*	Mean E Recall	Mean C Recall
unc6ia	-0.067 (7)	0.108 0.129	-0.181 to 0.046 -0.202 to 0.068	0.444 (6)	0.511 (1)
unc6ip	0.012 (5)	0.119 0.081	-0.113 to 0.136 -0.073 to 0.096	0.467 (4)	0.455 (5)

*The top value corresponds to the "sum.out" estimate (Over, 1997b) and the bottom value corresponds to the "sum-alt.out" estimate (Over, 1997c).

Table 2: Aspectual Precision Results. Rank among 10 interactive track runs is given in parentheses.

Run	Mean of Six E–C Estimates	Standard Deviation E–C*	95% Confidence Interval for Mean E–C*	Mean E Precision	Mean C Precision
unc6ia	-0.154 (10)	0.231 0.222	-0.396 to 0.089 -0.387 to 0.079	0.595 (10)	0.749 (8)
unc6ip	0.013 (3)	0.305 0.343	-0.307 to 0.333 -0.348 to 0.373	0.785 (5)	0.772 (6)

*The top value corresponds to the "sum.out" estimate (Over, 1997b) and the bottom value corresponds to the "sum-alt.out" estimate (Over, 1997c).

There were several problems with our searches that have a bearing on our results. First, searcher *irisa2i* did not save any documents during her three searches on IRIS. She put checkmarks next to aspects on her Searcher Worksheet though. We think she probably thought that marking a document as "Relevant" would be the same as saving it (see Figure 3). Hence, for her searches, any documents that she marked as relevant, we later marked as saved. Second, our system logs had no record of any evaluated documents (saved, relevant, etc.) for IRIS search *4_347* in run *unc6ia* and for IRIS search *6_307* in run *unc6ip*. However, on the Searcher Worksheets, 10 aspects were marked as saved for *4_347*, and 8 aspects were marked as saved for *6_307*. It is still unclear to us how these evaluated documents were lost. Third, with the exception of *irisp7i*, we neglected to tell the searchers that they should hit "Resubmit" in IRIS if they were viewing a ranking of documents at the end of the 20 minute time limit (see Figure 3). If they did not hit "Resubmit," any new documents that they had saved during that iteration would not be logged as saved by IRIS. Through examination of time logs, we conjecture that this may have had a negative impact on at least 1 search in *unc6ia* and at least 5 searches in *unc6ip*. (This problem may also have been a factor in searches *4_347* and *6_307*.) Finally, for search *1_326* in *unc6ia*, a bug in IRIS adversely affected the final ranking of documents.

In addition, two searchers (*irisa2i* and *irisa3i*) wrote on their Searcher Worksheets words from their queries instead of words used to describe aspects of the topic. For example, for Topic 339i the different aspects of the topic were the various drugs used to treat Alzheimer's Disease, but *irisa2i* wrote on her worksheet the words "alzheimer," "drug," "success," "treatment," "pharmaceutical," and "glaxo." The first four of these words had checkmarks next to them. An examination suggests that these searchers may have put checkmarks next to terms that had retrieved documents that they then saved. More investigation is needed to determine to what extent *irisa2i* and *irisa3i* misunderstood the goals of the interactive track experiment. Their values for aspectual recall are, in general, not worse than those for other UNC searchers.

3.3 Discussion

In the interactive track, mean E–C values are the principal measures used to determine whether differences exist among the experimental systems with respect to aspectual recall and aspectual precision. However, a high degree of overlap among the 95% confidence intervals for the ten runs makes any differences in the mean E–C values less meaningful. With respect to both aspectual recall and aspectual precision, the confidence interval for the best run overlaps the confidence interval for each of the nine other runs when either method for determining the confidence intervals is used (Over, 1997b, 1997c).

Regarding the performance of our two experimental systems, it is difficult to make a definitive statement because of the problems outlined earlier concerning the logging of saved documents. In addition, instructing searchers that they should utilize the full 20 minutes when searching on a topic may have also had a negative impact on our results. ZPRISE is faster than IRIS at retrieving documents, so, over the same time period, more search iterations can be conducted using ZPRISE. This may explain, in part, why run *unc6ia* had the best aspectual recall for ZPRISE out of the ten interactive track runs (see Table 1).

For run *unc6ia*, the searchers found that the collocations suggested by IRIS for possible addition to the initial query were helpful for most of the searches (see Table 3). Their responses are mixed for *unc6ip* (see Table 4).[3] As described previously, IRIS added to its term index those collocations that it determined to be statistically significant. It appears that at least one of the two words for most of these collocations occurred in a small number of documents. Hence, the collocations suggested by IRIS are perhaps most helpful with respect to topics that have specific aspects that are covered by few documents. For example, for the search on drugs used to treat asthma that was described previously, there were 10 suggested collocations that included the word "asthma" and the name of a drug used to treat it. However, a drawback of the method used to determine the statistically significant collocations is that many useful collocations that are not infrequent are not added to the term index. Accordingly, many collocations added to the query by searchers may not be in our index. This shortcoming was recognized too late in the process to correct in time for our TREC-6 runs.

Again, *unc6ia* employed the adaptive linear model, and *unc6ip* employed the probabilistic model. The searchers in *unc6ia* found relevance feedback in IRIS to be more beneficial to their sessions than the searchers in *unc6ip* (see Tables 3 and 4). In addition, on the exit questionnaires, searchers were asked which system they "liked the best" between IRIS and ZPRISE. The four searchers in *unc6ia* said "IRIS," whereas, in *unc6ip*, only one said "IRIS," while two said "ZPRISE" and one said she could not decide. It is difficult to generalize from two sample sets of only four searchers each. However, it is possible that the two relevance feedback models have different properties which may have influenced, in part, the searchers' responses.[4] Further investigation of the properties of the two models is needed.

[3] The last two questions in Tables 3 and 4 were taken directly from the "Post-Search Questionnaire" used by Rutgers (Belkin et al.) in the TREC-6 interactive track pre-experiment.

[4] However, any properties of the feedback models would not explain the differences between the two runs with respect to the searchers' attitudes toward the suggested collocations. Also, it is unclear how much of an influence on the responses was the fact that, for *unc6ip*, the marginally relevant documents were essentially treated like nonrelevant documents because the second term of Equation 8 was inadvertently ignored. Although further examination is needed, this may have had a relatively minor influence on the searchers' responses.

Table 3: For *unc6ia*, frequencies of answers to those questions on the IRIS Post-Search Questionnaire concerning the searcher's perceptions of the results of a search and of the impact on it by the suggested collocations and relevance feedback.

To what extent...	Not at all 1	2	Marginally 3	4	Extremely 5
were the suggested phrases for the initial iteration of the search helpful?		2		8	2
did relevance feedback help retrieve documents that cover new aspects?	2	1	3	5	1
did relevance feedback contribute in a positive way to the search?	2		2	7	1
did relevance feedback contribute in a negative way to the search?*	4	3	3	1	
are you satisfied with your search results?	2	3	3	3	1
are you confident that you identified all the possible aspects for this topic?	6	3		3	

*No answer was given for one search.

Table 4: For *unc6ip*, frequencies of answers to those questions on the IRIS Post-Search Questionnaire concerning the searcher's perceptions of the results of a search and of the impact on it by the suggested collocations and relevance feedback.

To what extent...	Not at all 1	2	Marginally 3	4	Extremely 5
were the suggested phrases for the initial iteration of the search helpful?	3	2	2	4	1
did relevance feedback help retrieve documents that cover new aspects?	4	3	4	1	
did relevance feedback contribute in a positive way to the search?	5	1	5	1	
did relevance feedback contribute in a negative way to the search?	2	7	2	1	
are you satisfied with your search results?	2	1	3	5	1
are you confident that you identified all the possible aspects for this topic?	6	2	3	1	

Although the searchers who used the adaptive linear model seemed to find relevance feedback more helpful than the searchers who used the probabilistic model, run *unc6ip* had better results in the interactive track than run *unc6ia* with respect to mean E–C values for both aspectual recall and aspectual precision. However, because the 95% confidence intervals overlap, the difference between the models could be explained by chance. In any case, further investigation of the relative retrieval effectiveness of the models is needed.

We were also interested in the number of documents declared relevant, marginally relevant, and nonrelevant by the searchers. These frequencies were also calculated for our manual adhoc run, so both sets of numbers will be presented in Section 5.

4 Manual Adhoc Runs

Run *unc6ma* was our Category B, manual adhoc run. Seven searchers searched for us using IRIS and the adaptive linear model. The searchers were either currently or had recently been graduate students in Library Science or

Information Science at UNC. Three searchers did 9 topics each, three did 6 topics each, and one did 5 topics. Most, if not all, of the searches took at least thirty minutes, and some took as long as an hour.

Table 5 contains performance values for the run, which are averaged over the 47 topics with at least one relevant FT document. In addition to overall average non-interpolated precision, the table includes average precision for the top 10, 20, and 30 documents retrieved. The last three values are included because we feel that (1) retrieval performance should be high for that set of documents that the typical searcher will evaluate and that (2) the typical searcher will usually not examine more than the top 30 documents. Finally, it should be noted that examining the final ranking of documents may not be the optimal way to evaluate an interactive retrieval session with multiple search iterations. The searcher may use one iteration of the search to retrieve documents that cover one aspect of the topic, and may use another iteration to retrieve documents that cover a different aspect. Our searchers, however, tried to produce the best final ranking of documents that they could.

Table 5: Performance measures for the manual, adhoc run (*unc6ma*). Values are averaged over the 47 topics with at least one relevant FT document.

Average non-interpolated precision	0.3663
Precision at 10 documents	0.4277
Precision at 20 documents	0.3309
Precision at 30 documents	0.2794

There were a few bugs in the system due to the fact that we were rushing to meet the TREC deadline. We are only aware of two topics that were affected by bugs. However, an in-depth investigation of the effect of bugs on our results is needed. One bug adversely affected the results for Topic 321. Another bug *only* affected Topic 303. The searcher appears to have been able to overcome the effects of this bug during the later iterations of the search.

After they had completed all of their searches, the searchers filled out an exit questionnaire. The frequencies of their answers to some of its questions are given in Table 6.[5] As in the interactive track, the searchers in general found that the suggested collocations were helpful. However, they did not think relevance feedback was as helpful as the searchers did for the interactive run *unc6ia* which also used the adaptive linear model. In fact, as shown in Table 6, a number of the manual searchers claimed that feedback contributed to the failure of searches. Oral and written comments by the searchers may explain, in part, the reason for this. Their comments indicate that there is still the problem of previously declared, nonrelevant documents floating to the top of the ranking. This is not a problem when the probabilistic model is used. This problem is particularly frustrating when the searcher wants to create a final query that will place all of the relevant documents before all of the nonrelevant ones. A simple solution to this problem though is to not include these previously declared, nonrelevant documents in the displayed ranking. (Not including them could be the "default" option in IRIS.)

5 Frequencies of Evaluated Documents

For both the interactive track and the manual adhoc task, we are interested in the number of documents declared relevant, marginally relevant, and nonrelevant by the searchers. Figures 6 and 7 display this information for the interactive track runs, *unc6ia* and *unc6ip*, respectively. The number of iterations in a search are displayed as well as the searcher's answer on the post-system questionnaire concerning the number of relevance levels that the searcher preferred to use.[6] For each searcher, the data are displayed in the order in which the topics were searched. (Again, the log of evaluated documents was lost for the search on topic 347i by *irisa4i* and for the search on topic 307i by *irisp6i*.) In each figure, the two searchers on the left searched on IRIS before they searched on ZPRISE, and the

[5] It should be stressed again that, in Tables 3 and 4 which refer to the interactive track searches, the questionnaires were filled out after *each search* on IRIS. In contrast, in Table 6, the questionnaire was filled out after all of the searcher's searches were completed.

[6] This question appears to have been misinterpreted by *irisa2i*, so her answer is not shown. Also, the options given the searchers were two levels, three levels, and four levels or higher (they could fill in a number). We did not think to include one level of relevance as an option (i.e., the only level would be "relevant").

two searchers on the right searched on ZPRISE before IRIS. The figures do not include any documents that were not logged due to the searcher's failure to hit "Resubmit."

Table 6: For *unc6ma*, frequencies of answers to those questions on the exit questionnaire concerning the searcher's perceptions of the results of the search and of the impact on it by the suggested collocations and relevance feedback. The exit questionnaire was filled out after all of the searcher's searchers were completed.

Rate the following...	**Not at all** 1	2	**Marginally** 3	4	**Extremely** 5
How confident are you that search outcomes were successful?			4	3	
Were the suggested phrases for the initial iteration of the search helpful?		1	3	1	2
Did relevance feedback contribute to the success of searches?			6	1	
Did relevance feedback contribute to the failure of searches?		1	4	2	
Were you satisfied with using three levels of relevance?	2		3	2	

Several points can be made about Figures 6 and 7. First, there is a high degree of variation among searchers with respect to the total number of documents evaluated as well as the percentage of documents assigned a given level of relevance. Second, for most of the searches, a high percentage of documents were declared nonrelevant. Third, only seven out of the twenty-two searches had documents declared marginally relevant.[7] Finally, an order effect can be detected. The searchers who searched on ZPRISE before IRIS generally seem to have evaluated fewer documents than those who searched on IRIS before ZPRISE. This difference is perhaps due to fatigue because the searches took place during one 3 ½ to 5 hour session. There is also some evidence of an order effect among the topics searched. In no case did the third topic searched have the most evaluated documents and in several cases it had the fewest.

Figures 8 through 10 show the number of documents declared relevant, marginally relevant, and nonrelevant by the manual adhoc searchers.[8] There are some similarities between Figures 8-10 for the adhoc task and Figures 6-7 for the interactive track even though the nature of the retrieval task was different between the two sets of runs. First, like the interactive track data, the adhoc task data show a high degree of searcher variation with respect to the total number of documents evaluated as well as the percentage of documents assigned a given level of relevance. (In addition, the adhoc task data show a high degree of searcher variation with respect to the number of iterations searched.) These results suggest that an operational IR system incorporating feedback needs to take into account such variation. Second, like the interactive track searches, many of the adhoc searches had a high percentage of documents declared nonrelevant. If this finding is substantiated by further research in a non-laboratory setting using users with real information needs, it would suggest that feedback retrieval systems perhaps should include a "nonrelevant" option as well as a "relevant" one.

[7]With respect to the *unc6ip* run, we need to further investigate whether discarding the second term in Equation 8 had an influence on the number of documents declared marginally relevant.

[8] The manual adhoc searchers indicated the number of relevance levels they preferred to use on the exit questionnaire. Also, like the interactive track searchers, the manual adhoc searchers may have hit "New Search" (see Figure 3) to restart a search using a new query. However, because the goal of the adhoc task was to produce a final query, the data shown in the figures are only for that sequence of iterations (possibly after "New Search" was entered) that produced that final query. Limiting the data in this way also makes the figures less cluttered with information about the number of iterations searched.

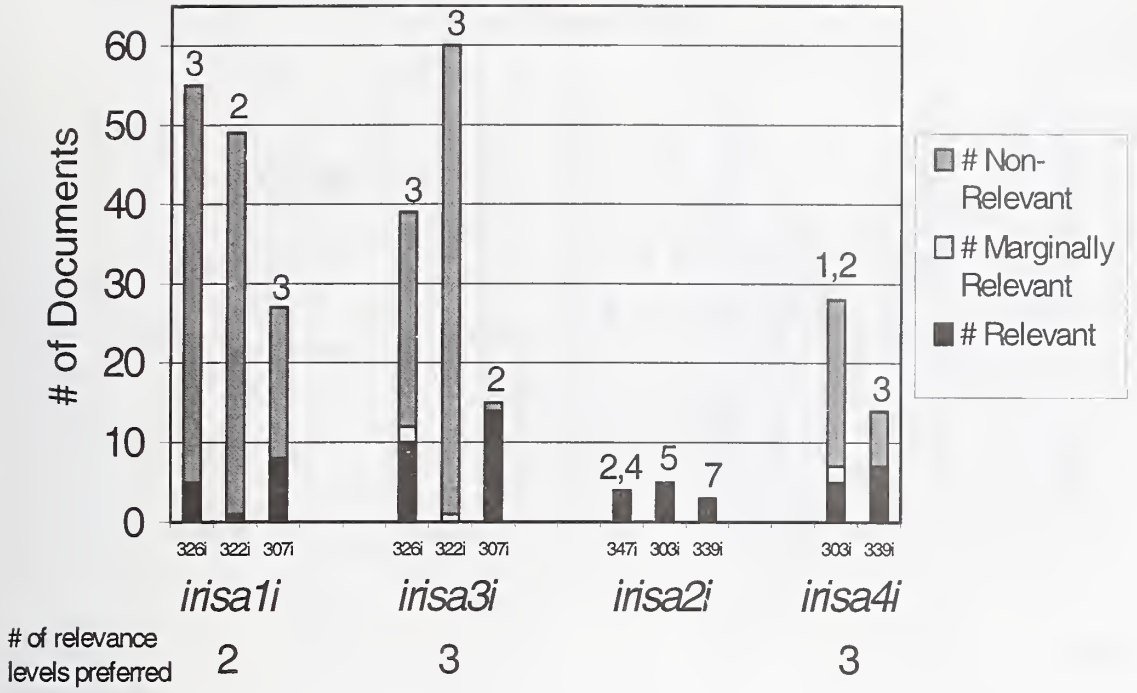


Figure 6: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers in run *unc6ia*. The number of iterations in a search is given above the appropriate column. (Two numbers are given if the searcher hit "New Search" to restart the search with a fresh query—see Figure 3.)

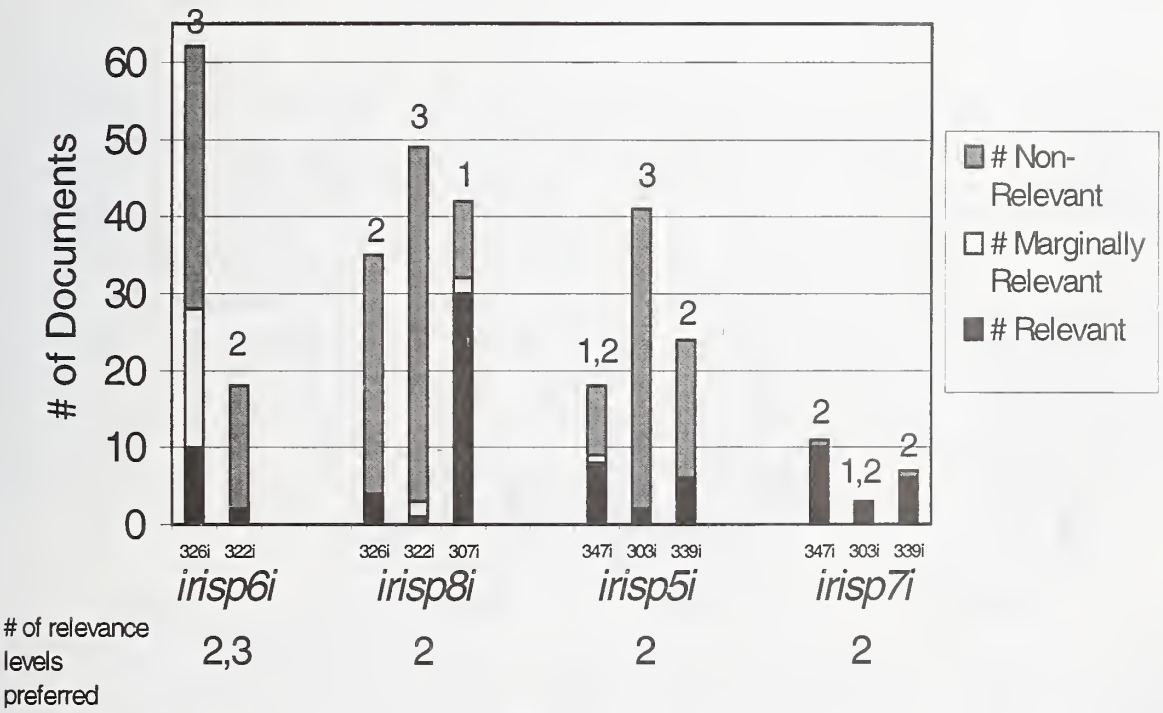


Figure 7: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers in run *unc6ia*. The number of iterations in a search is given above the appropriate column. (Two numbers are given if the searcher hit "New Search" to restart the search with a fresh query—see Figure 3.)

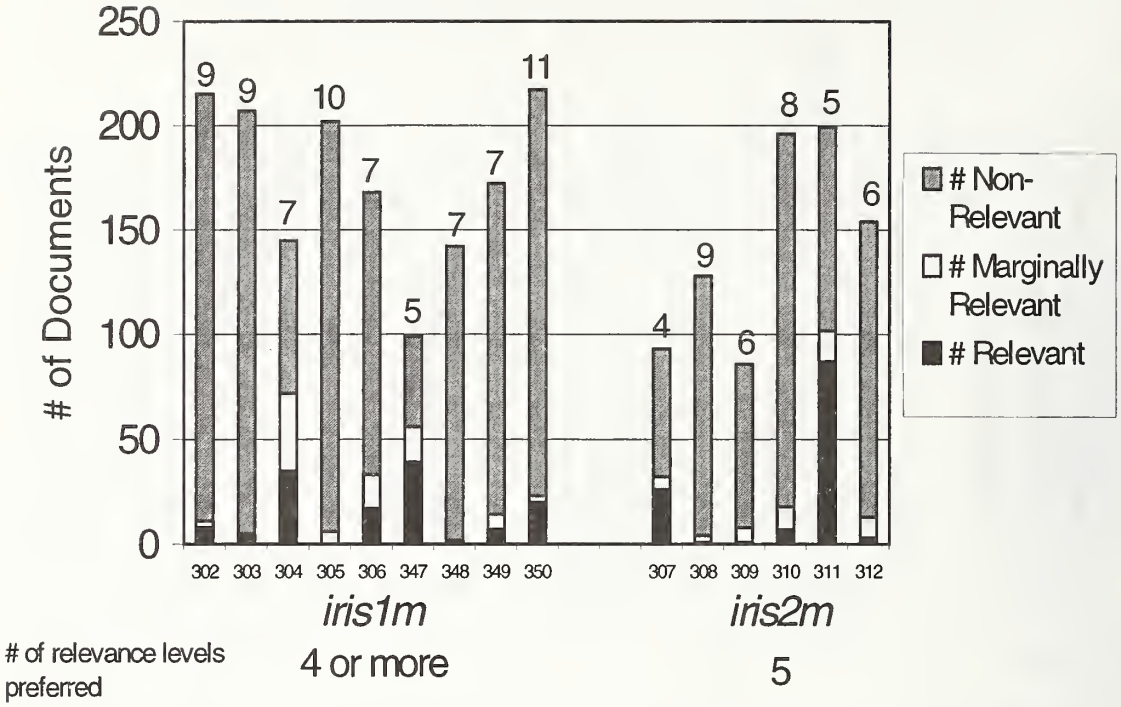


Figure 8: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers *iris1m* and *iris2m* in run *unc6ma*. The number of iterations in a search is given above the appropriate column.

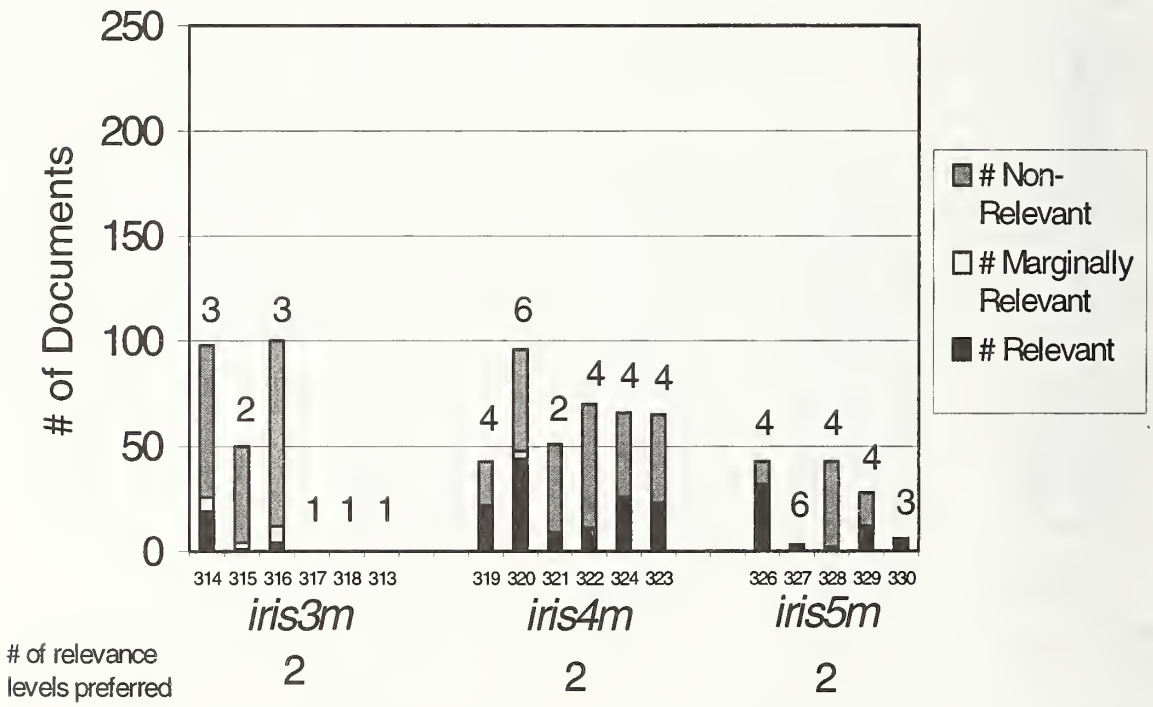


Figure 9: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers *iris3m*, *iris4m*, and *iris5m* in run *unc6ma*. The number of iterations in a search is given above the appropriate column.

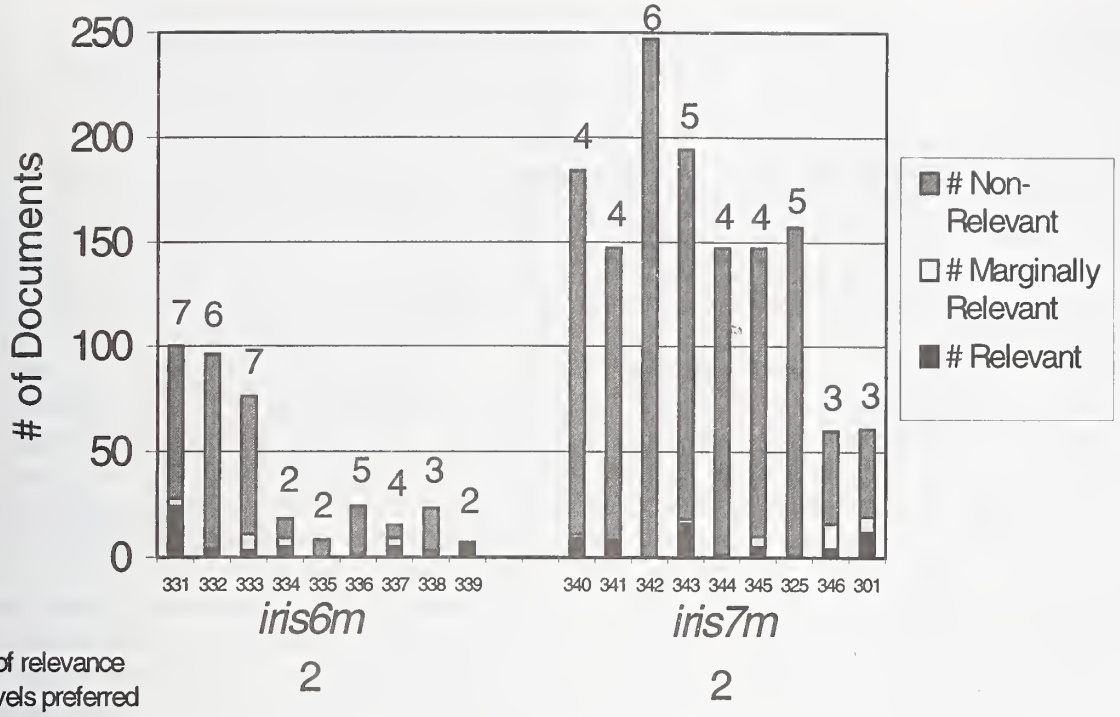


Figure 10: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers *iris6m* and *iris7m* in run *unc6ma*. The number of iterations in a search is given above the appropriate column.

There are also some differences between the figures for the adhoc task and the figures for the interactive track. First, the adhoc searchers did not have a time limit and sometimes took more than an hour per search. Hence, in contrast to the interactive track searchers, the adhoc searchers generally evaluated a larger number of documents and conducted a greater number of search iterations. Second, a larger number of the adhoc searches had documents that were declared marginally relevant. Thirty-two out of the 47 topics with evaluated documents had at least one document declared marginally relevant. If one excludes the searches of *iris4m* and *iris5m*, this ratio increases to 30 out of 36. This difference between the adhoc task and the interactive track is perhaps explained by the different nature of the two retrieval tasks as well as the greater amount of time spent by the adhoc searchers. Clearly, more research is needed to determine if feedback retrieval systems should include more than two levels of relevance. Our data has other evidence as well (besides the number of searches) concerning whether or not another level of relevance should be included. On the one hand, the figures for the adhoc task show that the number of marginally relevant documents for a search is often a high percentage of the number of relevant documents. On the other hand, a large majority of our TREC searchers said they preferred a binary relevance scale.

5 Automatic Adhoc Runs

Two Category B, automatic adhoc runs were submitted. Run *unc6aas* was the "short query" run, and run *unc6aal* was a "long query" run which utilized the description and narrative fields of the topic. The features of IRIS that do not require any human interaction with the system were employed in the automatic run. Accordingly, the initial query only consisted of single-word terms. No "suggested" collocations were added to it because that requires human judgment concerning which collocations would be the appropriate ones to add. The features that do not require human interaction were implemented the same way that they were in the interactive track and the manual adhoc task (e.g., the document term weights were Lnu weights with a slope of 0.3.)

Participants in previous TREC conferences have explored *top-document* feedback, where the top X documents as ranked by the original query are assumed to be relevant and then a feedback model produces a new query vector to re-rank the documents (e.g., Buckley et al., 1995). Of course, the success of this procedure is dependent on the quality of the initial query (Harman, 1996). We investigated using top-document feedback in which the adaptive linear and the probabilistic model are employed. We also tested using more than two iterations in this process. For example, let us assume that three iterations in all are utilized (one of which is the initial iteration). First, the top X documents from the initial ranking are assumed to be relevant and are used to produced a new query vector which re-ranks the documents. Then the top X documents from the second ranking are assumed to be relevant and are added to the training set in order to produce the final query vector which, in turn, produces the final ranking of documents. If the quality of the initial ranking is poor, using more than two iterations should have an adverse effect on retrieval performance.

We conducted our tests on a subset of the TREC-5 topics, and evaluated the results using the FT relevance judgments. The top X documents were assumed to be relevant and the next $100 - X$ were assumed to be nonrelevant. We varied the number of iterations and the "window size" (the value for X) in our tests. Table 7 has the results for the long query (title, description, and narrative) and for the adaptive linear and the probabilistic model. Firstly, the adaptive linear model performed much better than the probabilistic model, probably because nonrelevant documents are generally given lower ranks by the probabilistic model as opposed to the adaptive linear model. Accordingly, many of the $100 - X$ documents that are "officially" relevant will be given low ranks by the probabilistic model. Secondly, the window size of 5 performed marginally better than larger window sizes, perhaps due to the fact that, in general, the density of officially relevant documents is probably highest in that window. Thirdly, an unexpected result was that in some cases three iterations did better than two. Finally, the best result was for the adaptive linear model with a window size of 5 and with two iterations. These were the parameters that were used in the "long query" automatic run for TREC-6. However, it should be noted that the best result is only marginally better than the result for the initial iteration.

Table 7: Overall average non-interpolated precision for the long query.

Number of Iterations	Window Size					
	Adaptive Linear Model			Probabilistic Model		
	5	20	30	5	20	30
1	0.2082	0.2082	0.2082	0.2082	0.2082	0.2082
2	**0.2181**	0.2109	0.2060	0.0985	0.0836	0.0647
3	0.2176	0.2150	0.2097	0.1380	0.0463	0.0350

Table 8 contains the results of our testing using the 31 TREC-5 topics and the short query. Only the adaptive linear model was tested because of its superior performance using the long query. Results are similar to that for Table 7. Again, the best run was the adaptive linear model with a window size of 5 and with two iterations. These parameters were used in our TREC-6 short query run.

Table 8: Overall average non-interpolated precision for the short query and the adaptive linear model.

Number of Iterations	Window Size	
	5	20
1	0.1911	0.1911
2	**0.2020**	0.1905
3	0.2019	0.1846

Table 9 contains our official TREC-6 results for both *unc6aas* (the short query run) and *unc6aal* (the long query run). As to be expected, the manual run (see Table 5) did much better than the automatic runs, and the long query automatic run did better than the short query automatic run.

Table 9: Performance measures for the automatic adhoc task for both *unc6aas* (the short query run) and *unc6aal* (the long query run). Values are averaged over the 47 topics with at least one relevant FT document.

	unc6aas	*unc6aal*
Average non-interpolated precision	0.2167	0.2518
Precision at 10 documents	0.2766	0.3064
Precision at 20 documents	0.2138	0.2340
Precision at 30 documents	0.1738	0.1972

7 Future Research

We plan on improving IRIS over the next year or so. First, we may explore modifications to our method for determining statistically significant collocations. Other phrase generation methods may also be investigated. Second, we need to determine the number of levels of relevance with which the user evaluates documents for feedback. Third, we may explore other relevance feedback models that incorporate multiple levels of relevance. Fourth, we may compare using different starting vectors and values for α in the adaptive linear model (see Equation 4). Fifth, we may give the user more control over the feedback process by requiring her to explicitly add the new terms suggested by feedback. This is a model employed by Belkin et al. (1998) and others. Sixth, we may explore presenting the feedback terms to the user after each relevance evaluation of a document (Beaulieu & Gatford, 1998; Belkin et al., 1998) instead of waiting for the user to hit "Resubmit" after she has evaluated a number of documents. Seventh, we are currently working on an online interactive tutorial. Eighth, we are also working on ways to improve our interface. Ninth, we need to make IRIS faster. Finally, the most important thing we need to do is more testing with users with real information needs.

Acknowledgements – Chris Brannon and Scott Barker have given us invaluable computing support. Also, we would not have been able to do the interactive track and the manual adhoc task without our enthusiastic searchers: Sai Balu, Danielle Borasky, David Borasky, Linda Brett, Sally Fessler, Lisa Greenbaum, Wanda Gunther, Anne Langley, Karl Lietzan, Muzhgan Nazarova, Mark Rosso, Robin Shapiro, Lisa Smith, Rong Tang, and Lucinda Thompson. Finally, we would like to belatedly thank Judd Knott and Scott Barker for their indispensable computing support for our TREC-5 experiments.

References

Beaulieu, M. M., Gatford, M. J.. (1998) Interactive Okapi at TREC-6. *Proceedings of the Sixth Text REtrieval Conference*.

Belkin, N. J., Perez-Carballo, J., Cool, C., Lin, S., Park, S. Y., Rieh, S. Y., Savage, P., Sikora, C., Xie, H., & Allan, J. (1998). Rutgers' TREC-6 interactive track experience. *Proceedings of the Sixth Text REtrieval Conference*.

Berry-Rogghe, G. (1974). The computation of collocations and their relevance in lexical studies. In A. J. Aitken, R. W. Bailey, & N. Hamilton-Smith (Eds.), *The Computer and Literary Studies* (pp. 103-112). Edinburgh: Edinburgh University Press.

Bollmann, P., & Wong, S. K. M. (1987). Adaptive linear information retrieval models. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 157-163.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. In D. K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (NIST Spec. Publ. 500-225, pp. 69-80). Washington, DC: U.S. Government Printing Office.

Buckley, C., Singhal, A., & Mitra, M. (1997). Using query zoning and correlation within SMART: TREC 5. In E. M. Voorhees & D. K. Harman (Eds.), *The Fifth Text REtrieval Conference (TREC-5).*

Buckley, C., Singhal, A., Mitra, M., & Salton, G. (1996). New retrieval approaches using SMART: TREC 4. In D. K. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)* (NIST Spec. Publ. 500-236, pp. 25-48). Washington, DC: U.S. Government Printing Office.

Choueka, Y., Klein, S. T., & Neuwitz, E. (1983). Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. *Association for Literary and Linguistic Computing Journal*, 4(1), 34-38.

Firth, J. R. (1957). Modes of meaning. *Papers in Linguistics 1934-1951* (pp. 190-215). London: Oxford University Press.

Fishburn, P. C. (1970). *Utility theory for decision making*. New York: John Wiley & Sons.

Haas, S. W., & Losee, R. M. (1994). Looking into text windows: Their size and composition. *Information Processing and Management, 30*, 619-629.

Harman, D. (1996). Overview of the Fourth Text REtrieval Conference (TREC-4). In D. K. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)* (NIST Spec. Publ. 500-236, pp. 25-48). Washington, DC: U.S. Government Printing Office.

Krovetz, R. (1993). *Viewing morphology as an inference process.* Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 191-203.

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval,* 180-188.

Lee, J. H. (1996a). *Analyses of multiple evidence combination (Tech. Rep. No. IR-88).* Amherst: University of Massachusetts, Center for Intelligent Information Retrieval.

Lee, J. H. (1996b). *Combining multiple evidence from different relevance feedback methods (Tech. Rep. No. IR-87).* Amherst: University of Massachusetts, Center for Intelligent Information Retrieval.

Losee, R. M. (1994). Term dependence: Truncating the Bahadur Lazarsfeld expansion. *Information Processing and Management, 30*, 293-303.

Martin, W., Al, B., & van Sterkenburg, P. (1983). On the processing of a text corpus. In R. Hartmann (Ed.), *Lexicography: Principles and practice* (pp. 77-87). London: Academic Press, Inc.

Nilsson, N. J. (1965). *Learning machines: Foundations of trainable pattern-classifying systems*. New York: McGraw-Hill.

Over, P. (1997a). *TREC-6 interactive track specification*. At http://www-nlpir.nist.gov/~over/t6i/trec6spec. (Also see: "TREC-6 Interactive Track Report" in *Proceedings of the Sixth Text REtrieval Conference* (1998).)

Over, P. (1997b). *sum.out.* At http://www-nlpir.nist.gov/~over/t6i/sum.out. (Also see: "TREC-6 Interactive Track Report" in *Proceedings of the Sixth Text REtrieval Conference* (1998).)

Over, P. (1997c). *sum-alt.out.* At http://www-nlpir.nist.gov/~over/t6i/sum-alt.out. (Also see: "TREC-6 Interactive Track Report" in *Proceedings of the Sixth Text REtrieval Conference* (1998).)

Phillips, M. (1985). *Aspects of text structure*. Amsterdam: Elsevier Science Publishers.

Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science, 27,* 129-146.

Rocchio, J. J., Jr. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 313-323). Englewood Cliffs, NJ: Prentice-Hall.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science, 41*, 288-297.

Shaw, W. M., Jr. (1995). Term-relevance computations and perfect retrieval performance. *Information Processing & Management*, 31(4), 491-498.

Shaw, W. M., Jr. (1996). [Letter to the editor]. *Information Processing & Management*, 32, 636-637.

Shaw, W. M., Jr., Burgin, R., & Howell, P. (1997a). Performance standards and evaluations in IR test collections: Cluster-based retrieval models. *Information Processing & Management, 33*, 1-14.

Shaw, W. M., Jr., Burgin, R., & Howell, P. (1997b). Performance standards and evaluations in IR test collections: Vector-space and other retrieval models. *Information Processing & Management, 33*, 15-36.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 21-29.

Smadja, F., & McKeown, K. (1990). Automatically extracting and representing collocations for language generation. *Proceedings of the 28th Annual Meeting of the Association of Computational Linguistics*, pp. 252-259.

Sumner, R. G., Jr., & Shaw, W. M., Jr. (1997). An investigation of relevance feedback using adaptive linear and probabilistic models. In E. M. Voorhees & D. K. Harman (Eds.), *The Fifth Text REtrieval Conference (TREC-5).*

van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

van Rijsbergen, C. J., Harper, D. J., & Porter, M. F. (1981). The Selection of Good Search Terms, *Information Processing and Management, 17,* 77-91.

Wong, S. K. M., & Yao, Y. Y. (1990). Query formulation in linear retrieval models. *Journal of the American Society for Information Science, 41,* 334-341.

Wong, S. K. M., Yao, Y. Y., & Bollmann, P. (1988). Linear structure in information retrieval. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval,* 219-232.

Wong, S. K. M., Yao, Y. Y., Salton, G., & Buckley, C. (1991). Evaluation of an adaptive linear model. *Journal of the American Society for Information Science, 42,* 723-730.

Yu, C. T., Buckley, C., Lam, K., & Salton, G. (1983). A generalized term dependence model in information retrieval. *Information Technology: Research and Development, 2,* 129-154.

Yang, K., & Yang, K. (1997). Graded relevance in information retrieval. *Unpublished manuscript.*

Appendix: Characteristics of Interactive Track Searchers

Information about each searcher's background and searching experience was gathered from the pre-study questionnaires. All eight searchers had received at least a bachelor's degree. With respect to the *unc6ia* run, all four searchers had received a Master's in Library Science, were currently working as librarians, and were female. The number of years they had been "online searching" was 9, 2, ~15, and 3. With respect to the *unc6ip* run, one searcher was currently working on a Master's in Information Science, and another was working on hers in Library Science. A third searcher was a library technical assistant, and another was an administrative assistant for the University. Two of the searchers for the *unc6ip* run were female, and two were male. The number of years they had been online searching was 2, 2, 21, and 2. For both *unc6ia* and *unc6ip*, Tables A-1 and A-2 respectively show the frequencies of searchers' answers to questions regarding their searching experience. Many of these questions were directly taken from the "Pre-Search Questionnaire" used by Rutgers (Belkin et al.) in the TREC-6 interactive track pre-experiment.

Table A-1: For *unc6ia*, frequencies of searchers' answers to questions on the Pre-Study Questionnaire regarding searching experience.

How much experience have you had...	**None** 1	2	**Some** 3	4	**A great deal** 5
searching on computerized library catalogs					4
searching on CD ROM systems, e.g., Infotrac, Grolier			2	2	
searching on commercial online systems, e.g., Dialog, Lexis, BRS Afterdark			3	1	
searching on world wide web browsers, e.g., Mosaic, Netscape, Internet Explorer				1	3
searching on other systems	1			1*	
searching full-text databases		2	1	1	
searching in ranked-output information retrieval systems	2		1		1
searching in information retrieval systems that provide relevance feedback	3		1		
using a mouse-based interface				1	2
reading articles from the Financial Times	4				
reading articles from another business- or financial-oriented newpaper, magazine, or other publication (e.g., *The Wall Street Journal*, *BusinessWeek*)		2	1	1	

*OCLC, WorldCat

Table A-2: For *unc6ip*, frequencies of answers to questions on the Pre-Study Questionnaire regarding searching experience.

How much experience have you had...	**None** 1	2	**Some** 3	4	**A great deal** 5
searching on computerized library catalogs			1	2	1
searching on CD ROM systems, e.g., Infotrac, Grolier		1	3		
searching on commercial online systems, e.g., Dialog, Lexis, BRS Afterdark	1	1		1	1
searching on world wide web browsers, e.g., Mosaic, Netscape, Internet Explorer			1	1	2
searching on other systems	1		1*	1**	
searching full-text databases		1	2	1	
searching in ranked-output information retrieval systems	1	2	1		
searching in information retrieval systems that provide relevance feedback	1	2	1		
using a mouse-based interface	1			1	2
reading articles from the Financial Times	4				
reading articles from another business- or financial-oriented newpaper, magazine, or other publication (e.g., *The Wall Street Journal*, *BusinessWeek*)		1	2	1	

*library card catalogs, yellow pages, phone directory
**MEDLINE, UNCLE, OVID

Context-Based Statistical Sub-Spaces

TREC-6 Notebook Paper

Gregory B. Newby
University of North Carolina at Chapel Hill[*]

ABSTRACT

The technique described in this paper is similar to latent semantic indexing (LSI), although with some variation. Whereas LSI operates by performing a singular value decomposition (SVD) on a large term by document matrix of co-occurrence scores, the technique here operates by identifying eigenvectors and eigenvalues of a term by term matrix of correlation scores (derived from co-occurrence scores). The technique of identifying eigenvectors and eigenvalues from a correlation matrix is known as principal components analysis (PCA). Variations from the previous year's TREC work include work using sub-documents (paragraphs), and working with small sub-matrices consisting only of terms in a query, rather than working with all terms from the collection.

INTRODUCTION

The approach to TREC-6 described in this paper is based on principal components analysis, in which a term co-occurrence matrix is used as a basis for generating an "information space." Rather than pursuing a model that applies a single information space for all queries, this year's TREC effort builds a custom information space for each query. This "context based" approach is intended to better distinguish among documents which possess terms from the query than an approach that simply folds all terms and queries into a much larger generic information space.

Retrieval from the space is typical of vector-space and related methods, in that queries are represented as pseudo-documents and then document similarity scores are computed between each document and the query/pseudo-document. The two main differences are that term and document vectors are not normalized, and that the geometric distance measure is used, rather than a cosine, dot product, or other measure.

The space-building process is as follows:

1. Identify a list of "good" words (words of interest) for a collection.
2. Count the co-occurrence of each good word with other good words across all documents in the collection. This is the symmetric term by term co-occurrence matrix for the collection.

[*] Address: CB-3360 Manning Hall; Chapel Hill; NC; 27599-3360; USA. Email gbnewby@ils.unc.edu.

3. For a query, identify the query's unique "good" words. Build a co-occurrence matrix based on the query context by extracting only these "good" term pairs from the collection co-occurrence matrix.
4. Principal components analysis (PCA) is performed on the query correlation matrix (generated from the query co-occurrence matrix) to identify eigenvectors and eigenvalues. The eigenvectors are the basis of the multidimensional information space, in which each term has a known numeric (metric) relationship to all other terms. The eigenvectors are used as coordinates of the query words in a multidimensional space.
5. Documents are located at the geometric center of the terms they contain.
6. Queries are represented as pseudo-document by also locating them at the geometric center of the terms they contain.
7. Retrieval proceeds by determining which documents are closest to the query, with closer documents interpreted as "more relevant."

Three TREC-6 tasks were performed: the adhoc task, the routing task and the filtering task. All tasks were performed on the "Category A" dataset (the full dataset), and all used only "automatic" query construction and retrieval. Query expansion was not used, nor was term weighting.

Comparable techniques were used by the author in TREC-5 (Newby, 1996). Additional techniques employed beyond the TREC-5 methods are:

1. For all three tasks, a query-specific context subspace was generated. From all original "good" words, only those words in the query were used. This resulted in a far smaller co-occurrence matrix than for TREC-5, when the same 1900 by 1900 term co-occurrence matrix was used for all queries.
2. For the filtering task, sub-documents were used, rather than entire documents. Each sub-document was a paragraph from the original document.

The most important factor lacking in the approach described here appears to be term weighting. Due to a complete absence of term weighting, the results presented here indicate that documents were retrieved based on the presence of some query terms, but the terms were not necessarily conceptually "important" to the query. Specifically, query terms with high collection frequencies (*tf*) would result in retrieval of many documents with these high-frequency terms. Meanwhile, the more important query terms, which generally had lower *tf*'s, added relatively fewer documents to the retrieved set. Current efforts are being directed at identifying useful term weighting schemes. Because the term by term co-occurrence score, which is the basis for the technique here, is derived from two separate terms, it is not yet clear whether traditional *tf / idf* weighting will be appropriate.

The remainder of this document discusses the outcomes of each TREC-6 task. Following a section on visualization, a concluding section summarizes the work to date for TREC, and identifies the most important areas for continued development.

THE ROUTING TASK

The routing task was based on 50 queries with pre-existing relevance judgments. The data were new data from the FBIS (Foreign Broadcast Information Service), a total of 120,654 documents. Topics used were: 1 3 4 5 6 11 12 23 24 44 54 58 62 77 78 82 94 95 100 108 111 114 118 119 123 125 126 128 142 148 154 161 173 180 185 187 189 192 194 202 228 240 282 10001 10002 10003 and 10004.

21,494 "good" words were identified for this task. The words were derived from various lists of dictionary terms, places and proper names. In addition, all terms from the 50 queries were put on the "good" words list. 21,494 is a post-stemming count – a Porter stemming algorithm was adapted from Frakes & Baeza-Yates (1992). A stoplist based on the SMART team's list was utilized (596 words).

Each of the 120,654 documents was analyzed for its contribution to a 21,494 by 21,494 co-occurrence matrix. For each document:

- All unique (stemmed) terms in the document were identified.
- Only those terms on the "good" words list were kept.
- For each term pair (**[N * (N-1)] / 2** pairs per document), the co-occurrence score in the full co-occurrence matrix was incremented by one.

The frequency of terms within documents (*df*) was not taken into account. The co-occurrence matrix thus counts the number of documents with each term pair, not the raw frequency of term pairs within all documents. The possible range of co-occurrence scores for the 230,996,018 ([21494^2] / 2) term pairs is 0 to 120,654 (the identity row was pre-defined as 1, in order to prevent zero variance for any rows in which all other scores were zeroes).

The number of pre-judged relevant documents from prior years for the 50 queries ranged from 18 to 2661, with a median of 131. This work did not take judgments of non-relevant documents from prior years into account.

The retrieval process was as described in the Introduction, above. Specifically:

1. 21,494 "good" stemmed words were identified by culling various word lists and pre-selecting all non stoplist terms from the 50 queries.
2. The 120,654 FBIS documents were used to build a 21,494 by 21,494 co-occurrence matrix.
3. A context co-occurrence matrix of *only those terms from the query* was extracted from the larger 21,494 by 21,494 co-occurrence matrix. Matrices for queries ranged in size from 3 by 3 to 106 by 106, with a median of 29 by 29.

4. Principal components analysis was performed on the context matrix, resulting in a multidimensional information space.
5. All FBIS documents with at least 25% of the query terms were located in the space. Documents that didn't meet the 25% cutoff were assumed to be non-relevant.
6. All pre-judged documents from prior TRECs were also located in the space.
7. Results were produced by retrieving the closest FBIS documents to the query and to the pre-judged documents.

Steps 6 and 7 were the points of departure from the adhoc task (below). Because no basis was developed for deciding which of the pre-judged documents was most important, or for determining a minimum cutoff value for closeness, a round-robin approach was used. For example, if 49 pre-judged documents were available from prior TRECs for a given query, these 49 plus the query were alternatively used to retrieve the next closest document (up to the desired set size of 1000 documents per query). In this case, the result would be the 20 closest documents to each of these 50 targets – but intermixed. This is not a very good criterion, but because the absolute metric values for each sub-matrix were different, there was no single cutoff value to use across queries.

The exceptions to this round-robin approach occurred when either there were more than 200 documents pre-judged as relevant, or when fewer than 1000 FBIS documents total met the 25% criteria. In the first case, time constraints prevented dealing with more than 200 pre-judged documents (this eliminated some pre-judged documents for almost half of the queries). In the second case, sometimes fewer than 1000 FBIS documents had 25% of the query terms, especially for long queries, so fewer than 1000 FBIS documents could be ranked for retrieval.

A second set of routing results was submitted (not for assessment) in which only those documents closest to the query were retrieved. In other words, in which none of the pre-judged documents were taken into account. This makes this set of results more like those of an adhoc task. The goal was to make some approximation to the prior year's TREC-5 effort, but with using context-specific subspaces rather than one larger space for all queries. In fact, results from this method do appear to be better: exact precision scores were higher, and the overall average percentage of relevant documents retrieved per query was higher. Further analysis will identify additional trends between the TREC-5 and TREC-6 results.

Across the pooled cohort group, an average of 146.2 (Standard Deviation = 180.9) relevant documents per query were found, with a range of 4 to 890. The first set of routing results had an average of 56.4 (SD 77.7), with a range of 0 to 388. The second set had an average of 18 (SD 24), with a range of 0 to 113.

Across all queries, the first set of routing results succeeded in identifying 36% of all pooled relevant documents for the query (SD 20%). The second set retrieved 11% (SD 9.6%). In both cases, the Pearson correlation between the number of relevant documents

was strongly correlated ($r = .88$; $p < .0001$) with the number of pooled relevant documents.

Exact precision scores for the first set ranged from 0.0 to .52, with a median of .10. For the second set, scores ranged from 0.0 to .12, with a median of .01.

The overall interpretation of these results is that the routing approach described here is effective at "query by example," in the case of the first retrieved set, where existing relevant documents are used as surrogate queries, and similar documents are found. Further investigation may help to discover whether the main benefit of query by example is the similar lengths of surrogate query documents and the real documents (versus shorter "real" queries), or other factors.

Additional investigation should be made of the performance of the techniques described here for clustering previously judged documents. Measuring the presence and densities of relevant or non-relevant clusters could be extremely effective for identifying useful regions in the information space for the retrieval of new documents.

THE ADHOC TASK

The adhoc task was based on 50 queries numbered 301-350. The data for the task were from discs 4 and 5: a total of 555,871 documents.

Adhoc Document Counts

Congressional Record	CR	27716
Los Angeles Times	LA	131896
Foreign Broadcast Information Service	FBIS	130471
Federal Review	FR	55630
Financial Times	FT	210158

The adhoc task made use of full documents (not sub-documents). Two sets of results were submitted. The first was created the same way as the second routing set described above. That is, the query was located in the multidimensional subspace, and the 1000 closest adhoc documents to the query were retrieved. (If fewer than 1000 adhoc documents had 25% or more of the query terms, only those documents that made the cutoff were retrieved.)

A second non-assessed set of results utilized only the Description field of the query. This resulted in very short queries, ranging from 2 to 10 "good" terms each. Because no query expansion was applied, results were uneven. Query 316, for example, had the description "A look at the roots and prevalence of polygamy." Only *roots*, *prevalence* and *polygamy* were on the list of "good" words, but *polygamy* did not occur in any of the 555,871 documents! The same 25% cutoff value was employed for this results set.

Across all 50 queries, a pooled total of 72,270 documents were judged based on submissions from all TREC participants (53,650 unique documents). On average, 92.2 (Standard Deviation 103.1) relevant documents per query were identified, with a range from 3 to 474

The first set of results (utilizing full queries) yielded an average of 5.5 (SD 8.1) relevant documents per query, with a range from 0 to 35. A correlation of -.53 ($p < .0001$) between the total pooled relevant documents per query and the number of relevant documents from the first set was found. This indicates that the approach taken was less effective when many relevant documents were found by the TREC cohort. Or, inversely, that the approach was more effective for more "difficult" queries, in which fewer relevant documents were found by the cohort. However, the correlation was practically identical (-.52, $p < .0001$) for the pooled cohort group, indicating a likelihood that the system described here was typical in this regard.

A correlation of .80 ($p < .0001$) between the number of relevant documents in the first set and the total number of pooled relevant documents indicates, as expected, that the number of relevant documents in the set increases with the total number of relevant documents.

Recall and precision scores for this set were poor, with exact precision scores ranging from 0.0 to .08, with a median of 0.0. Overall, this set identified a per-query average of 5% of the pooled relevant documents, with a range from 0% to 24% (SD 5.6%).

The second set had somewhat better performance figures, but the soundest interpretation appears to be that the improvement was simply due to the greatly decreased number of query terms. By seeking a minimum of 25% of query terms per retrieved document (before locating the document in the information space and ranking it for retrieval), a somewhat better response set might be expected simply by this Boolean approximation to a first cut. However, greater than 1000 documents were generated by this cut for each query – indicating that even if the Boolean approximation increases effectiveness, the information space distance ranking could still be an important component of the improvement over the first set's results. Further investigation of the role of the Boolean approximation is being made by the author.

For the second set, the overall statistics for the number of pooled judgments and relevant documents found is the same as for the first set (both set's statistics were derived from the overall TREC cohort group). But the mean number of relevant documents per query in this set was 9.92 (SD 19.3), versus 5.5 for the first set. The range was from 0 to 125.

The correlation between the number of relevant documents in the second set and the total number of pooled relevant documents was .67 ($p < .0001$), while the correlation between the number of relevant documents found by the cohort group and the number found in this set was not significant (at $\alpha = .05$). These scores indicate that the approach taken here exhibited a different pattern, overall, than the cohort for the relation between the

number of pooled relevant documents and the number of relevant documents retrieved for a given query. This pattern was also different than for the first retrieved set.

While the recall and precision scores for this second adhoc set were only slightly less poor than for the first set, the exact precision scores were somewhat improved with a range from 0.0 to .18, with a median of .01. Overall, this set identified a per-query average of 9.1% of the pooled relevant documents, with a range from 0% to 35.5%.

In summary, the adhoc results generated by the approach described here were not outstanding. Rather than attribute the enhanced performance of the second set (with short queries) to the information space approach, a simpler interpretation is that the 25% cutoff yielded a better sub-set of documents to locate in the information space, thereby yielding better results.

The overall pattern of adhoc results indicates the context-based information space approach may have some potential for usefulness, but not without additional work. The first set's pattern of retrieving 5% of relevant documents is only somewhat better than the 2% expected by chance through random selection. But when specific queries are examined, there appears to be more promise: some queries in both sets had reasonable performance measures (i.e., close to or above the median). Based on the relatively superior performance of the second set, it appears wise to investigate methods for term weighting or automatic query processing (expansion or term removal) in order to maximize the effectiveness of the set of documents eligible for ranking and retrieval.

THE FILTERING TASK

The filtering task was based on the same 50 queries and data as the routing task (120,654 FBIS documents). These documents were broken into sub-documents by inserting paragraph tags into the original documents whenever a line started with two spaces. (Unfortunately, a number of documents did not meet this standard, notably some transcriptions of interviews. These documents were treated as one long sub-document.)

This process resulted in a total of 1,909,729 sub-documents. Paragraphs with no "good" words were ignored and not counted. The number of sub-documents per document ranged from 1 to 4726.

The rationale for working with sub-documents is twofold. First is an effort to decrease the size of "documents" (that is, sub-documents) to be retrieved. Because the size range for full documents is great, this could lead to less variance in the size ranges for sub-documents – hopefully preventing an uneven likelihood of longer documents being retrieved versus shorter documents. The second component of the rationale is to simply investigate the applicability of the multidimensional information space approach for identifying useful sub-documents.

Otherwise, the retrieval process was identical to the process for the routing task, but with differences as to the final selection criteria for which documents were included in the retrieved set. Unlike the routing and adhoc tasks, in which a ranked set of retrieved documents is presented, the filtering task documents were not ranked. Instead, a binary relevance judgment about whether to accept or reject each FBIS document was to be made independently of judgments for other documents.

Multiple evaluation criteria, F1 and F2, were applied to the filtering task. Essentially, F1 penalizes more heavily for non-relevant documents being retrieved (favoring high precision), while F2 had a lesser penalty for non-relevant documents but an added penalty for not retrieving known relevant documents (balancing high precision with high recall).

For the F1 evaluation criteria, a conservative distance value for a cutoff was needed. This cutoff would determine a hypersphere around each query within which all documents would be retrieved. Because the metric of each query space (that is, the range of eigenvectors) was different from other spaces, there was no acontextual method for determining a cutoff value. In other words, it was not possible to choose a value (such as "10 units") which would be suitable for all query information spaces.

The conservative cutoff was chosen as the *smallest* of the distances from all pre-judged documents to the query. Any FBIS document that was closer to the query than this distance was retrieved. This resulted in fewer than the maximum of 1000 documents being submitted for all but 5 queries (240, 194, 180, 173 and 125), with a low of 1 document being submitted (query 10002) and a median of 158. Unlike routing, distances of documents to previously judged relevant documents were not utilized. It will be interesting, in the future, to see the effects of taking the same exact approach for routing and filtering, with the only difference being the use of full documents versus sub-documents.

The retrieved set for function F1 was poor. Of the 47 queries, the lowest score was achieved on 22 queries (46.8%), and none of the scores exceeded the median. In all but three of the lowest achieved scores, the number of FBIS document retrieved exceeded the pooled total number of relevant documents for the FBIS collection. In all, the 17 queries for which *fewer* than the pooled total number of relevant document were retrieved had higher F1 scores than the 30 queries for which *greater* than the pooled number were retrieved. This points to a problem of not only having few relevant documents in the retrieved set, but also being unable to discard non-relevant documents.

The retrieved set for function F2 was considerably better than the set for F1. This set utilized a 25% cutoff for the minimum number of query terms per document. Because of the relatively high cutoff, fewer than 1000 documents were eligible for being assigned a location in the information space. The range of eligible documents was from 0 (queries 23 and 148) to 843 (query 202), with a median of 16.

In effect, the F2 set was produced by Boolean probabilistic methods, in that results were not ranked (a ranked set is mentioned below). One query achieved the highest F2 score (query 23, with no documents retrieved) and 8 queries (17%) were close to or above the median. Six queries (12%) achieved the lowest score. For 37 queries (79%), the number of documents retrieved was lower than the total number of relevant documents. The results of F1 and F2 are indicative of the merits of retrieving very small sets for the filtering task.

Two additional sets of filtering results were delivered based on ranking of documents, rather than on binary filtering. The first set, based on F1, simply included the closest 1000 documents per query. In other words, the same results set as would have been submitted for the routing task. In practice, the results were not the same, because the 25% cutoff was not applied. Instead, FBIS documents with *any number greater than 1* of the query terms were located in the information space and eligible for ranked retrieval.

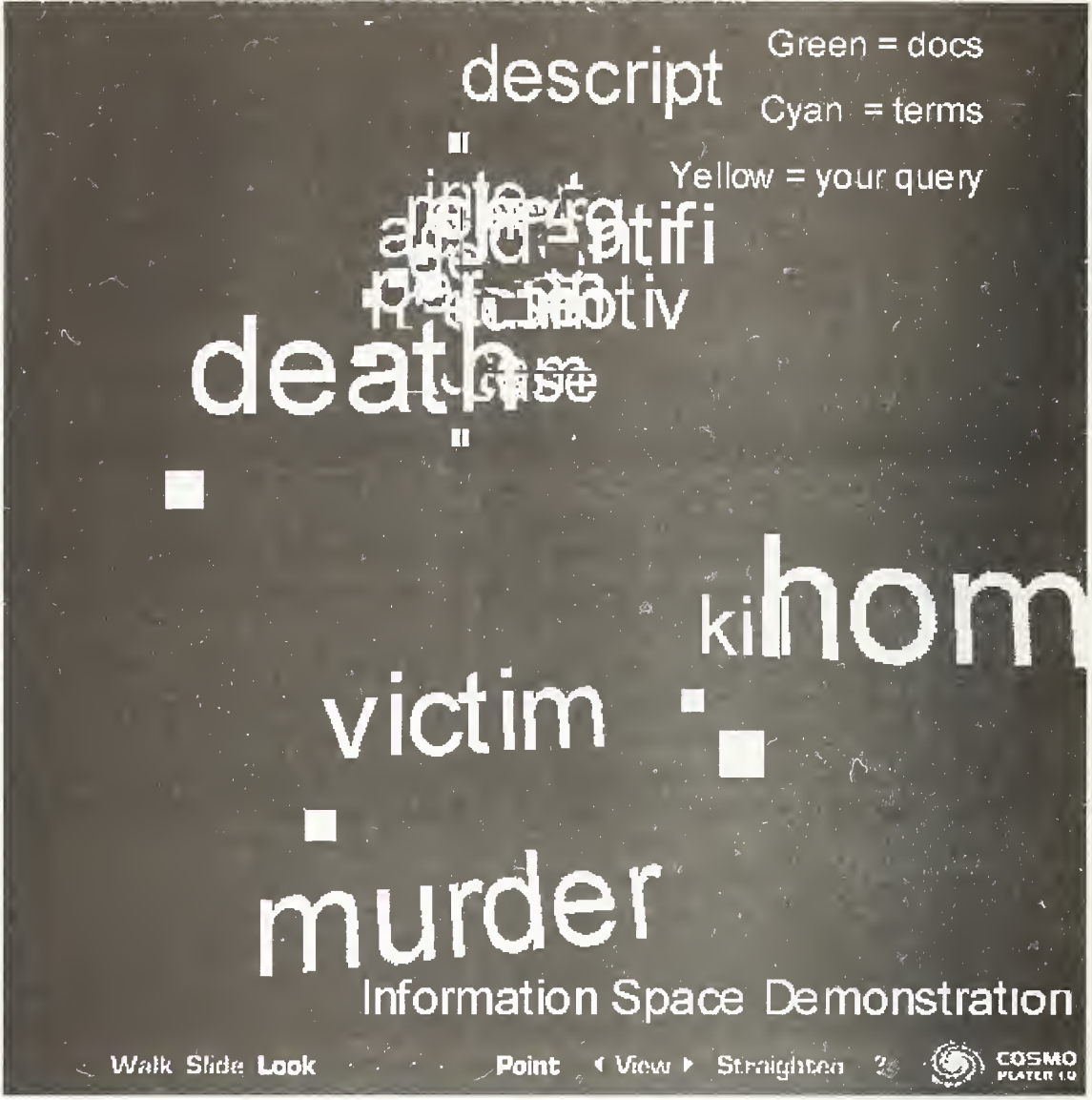
The second ranked additional set, based on F2, applied a 10% cutoff, but was otherwise comparable to the additional set based on F1. For the main F1 and F2 sets, as well as the additional ranked set for F1, the recall and precision scores were very low. Typically, an average of only 3% to 6% of the pooled relevant documents were retrieved per query. Exact precision was poor for the F1 sets (with maximum scores of .10), but better for the F2 sets (with a maximum of .44 for the unranked set, and .29 for the ranked set).

The filtering results are, in general, comparable to the routing results. The ability of the information space technique to identify non-relevant documents, in these experiments, was created more by the Boolean inclusion of documents based on query terms than on a particular cutoff distance from the query to the document.

The use of sub-documents did yield better recall and precision scores than were found for the routing task, which used entire documents. Further analysis will be completed to determine whether the retrieved sets of documents is appreciably different from sets retrieved by other TREC groups or by using full document information space methods. Numerically, the retrieved sets appear to be very different: less than 1/3 of the documents retrieved in routing were also retrieved in filtering, and vice-versa.

VISUALIZATION

Throughout the adhoc, routing and filtering tasks, retrieval was from a series of programs operating in batch mode. That is, each document collection, then every query, was run through a series of steps without human intervention. This section makes brief mention of one further application of the information space techniques described here: the visual interface.



The routing, adhoc and filtering task demonstrated capability of the context-based information space approach described here for clustering documents based on similarity, then retrieving documents based on their similarity to a query. Given that PCA extracts the largest eigenvalues first, it is possible to view the first three dimensions of the information space such that it is visually indicative of the overall multidimensional space. From 25% to 75% of the variance of the entire co-occurrence matrix is accounted for by these first three eigenvalues, enabling a reasonably accurate view.

The figure shows a VRML fly-through model of Query 189, accessed through the Cosmo Player plug-in to Netscape version 4. Query terms are visible; the query is towards the center of the space. The closest 100 documents to the query are displayed as dark squares, and may be clicked to retrieve the document.

The author has developed a Web-based interface to the entire information space-building system, with the capability of producing such a VRML world automatically. Because the process takes from 5 to 20 minutes, depending on the length of the query, this is not yet

an alternative to existing real-time interfaces. The author is performing usability testing on the VRML interface, as well as an OpenGL version which functions with MSWindows and X-Windows.

DISCUSSION

These results for the adhoc, routing and filtering tasks show good improvement over the information space approach used in TREC-5. However, the results are far from outstanding. Further work is needed to develop term weighting techniques and methods for automatic query processing (expansion and truncation). The use of sub-documents seems promising, and would be especially interesting using manual relevance feedback or otherwise extracting relevant *passages* from previously judged relevant *documents*.

Some anomalies of the information space approach remain. The variability of the basic scale of the space (e.g., that the average inter-document distance varies greatly in different contexts) is clearly based partially on the number of query terms, but the nature of the relationship is not clear.

Visualization of information space is powerful for evaluating the clustering of documents, query terms and queries. Visual feedback systems are being developed that may be suitable for manual, versus automatic, participation in future experiments. Given the relatively unimpressive results for the three TREC-6 tasks described here, it may be that one of the main roles of the information space approach will be to create visual, spatial and navigable spaces based on response sets derived from traditional approaches. For example, a traditional Boolean, probabilistic or vector system might be used to generate a response set of potentially relevant documents for further processing and visualization using the information space approach.

Ongoing work includes further evaluation of the TREC-6 results to identify trends in the types of documents that are retrieved (length, presence of particular terms, etc.). In addition, queries with greater or lesser success will be examined to ascertain the relation between different types of queries and the effectiveness of the information space approach.

REFERENCES

Frakes, Willaim B. & Baeza-Yates, Ricardo, Eds. 1992. Information Retrieval: Data Structures and Algorithms. Englewood Cliffs, NJ: Prentice-Hall.

Newby, Gregory B. (1997). "Metric multidimensional information space." In Harman, Donna (Ed.). Proceedings of the TREC Conference, volume 5. Gaithersburg, MD: NIST.

THE THISL SPOKEN DOCUMENT RETRIEVAL SYSTEM

Dave Abberley (1), Steve Renals (1), Gary Cook (2) and Tony Robinson (2,3)

(1) Department of Computer Science, University of Sheffield, UK
(2) Department of Engineering, University of Cambridge, UK
(3) SoftSound, UK

1. INTRODUCTION

The THISL spoken document retrieval system is based on the ABBOT Large Vocabulary Continuous Speech Recognition (LVCSR) system developed by Cambridge University, Sheffield University and SoftSound, and uses PRISE (NIST) for indexing and retrieval. We participated in full SDR mode.

Our approach was to transcribe the spoken documents at the word level using ABBOT, indexing the resulting text transcriptions using PRISE. The LVCSR system uses a recurrent network-based acoustic model (with no adaptation to different conditions) trained on the 50 hour Broadcast News training set, a 65,000 word vocabulary and a trigram language model derived from Broadcast News text. Words in queries which were out-of-vocabulary (OOV) were word spotted at query time (utilizing the posterior phone probabilities output by the acoustic model), added to the transcriptions of the relevant documents and the collection was then re-indexed. We generated pronunciations at run-time for OOV words using the Festival TTS system (University of Edinburgh).

Our key aims in this evaluation were to produce a complete system for the SDR task, to investigate the effect of a word error rate of 30-50% on retrieval performance and to investigate the integration of LVCSR and word spotting in a retrieval task. To achieve this we performed four basic experiments indexing on: transcribed text; IBM (baseline recognizer) SRT files; ABBOT SRT files; and ABBOT SRT files combined with word spotting of OOV words in the query.

This evaluation provided a stress test for our LVCSR system. In particular we developed our decoding algorithm and software to operate in a more "online mode". The result of this was the ability to decode arbitrarily long passages without segmentation into "utterances". When indexing, acoustic model computation required around 3.5× real time on a Sun Ultra 1/170, and lexical search required around 2.5× real time. At query time the word spotting component ran in about 0.25× real time per document per query.

This work was supported by ESPRIT Long Term Research Projects SPRACH (20077) and THISL (23495).

2. SYSTEM ARCHITECTURE

The outline of the basic THISL system is illustrated in figure 1. The ABBOT LVCSR system was used to provide approximate transcriptions of the audio documents so that the task could be treated as one of text retrieval. Since the current ABBOT system uses a finite vocabulary of around 65 000 words, a query-time wordspotter was incorporated to allow words that were OOV with respect to the LVCSR system to be retrieved.

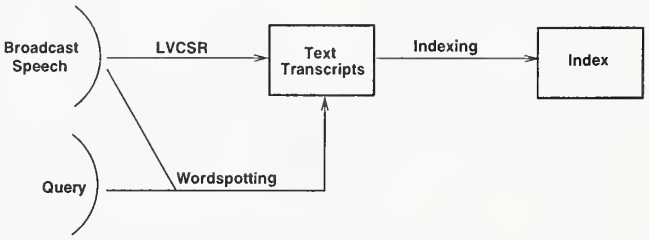


Figure 1: The indexing portion of the THISL Spoken Document Retrieval system used in TREC-6.

3. THE ABBOT LVCSR SYSTEM

ABBOT is a hybrid connectionist/HMM system [1] that differs from traditional HMMs in that the posterior probability of each phone given the acoustic data is directly estimated at each frame, rather than the likelihood of a phone (or state) model generating the data. This posterior probability estimation is achieved by using a connectionist network trained as a phone classifier. In the ABBOT system, a recurrent network [2] is used as the acoustic model (figure 2). Direct estimation of the posterior probability distribution using a connectionist network is attractive since fewer parameters are required for the connectionist model (the posterior distribution is typically less complex than the likelihood) and connectionist architectures make very few assumptions on the form of the distribution. Additionally, this approach allows for an efficient search algorithm that uses a posterior

probability-based pruning (section 3.3) [3] and is able to provide useful acoustic confidence measures [4].

Since the likelihood is required in the decoding process, the posterior is converted to a scaled likelihood, $L(x;q)$. This may be computed by dividing the posterior probability estimate of phone (or HMM state) q given the data x, by the class prior $P(q)$ estimated as the relative frequency in the training data:

$$L(x;q) = \frac{P(q|x)}{P(q)} = \frac{p(x|q)}{p(x)} . \quad (1)$$

The assumptions underlying this acoustic model are discussed in detail in [1, 5].

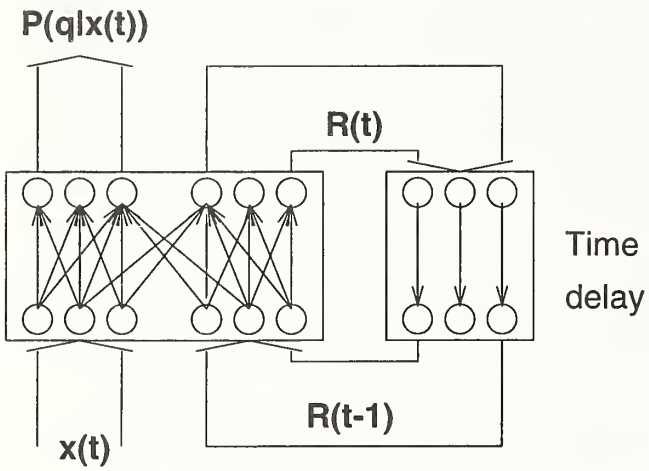


Figure 2: Recurrent network architecture used for acoustic modelling in the hybrid connectionist/HMM approach.

3.1. Acoustic Model

The acoustic model used in the THISL system consisted of two recurrent networks with 53 context-independent phone classes (plus silence). One network estimated the phone posterior probability distribution for each frame given a sequence of 12th order perceptual linear prediction features [6]. The other network performed the same distribution estimation with features presented in reverse order (since recurrent networks are time-asymmetric) and the two probability estimates were averaged in the log domain.

The context independent probability estimates ($P(q|x)$) were combined with a context class posterior probability $P(c|q,x)$, where c is an acoustic context class, to give the joint posterior probability of context class and phone class, $P(q,c|x) = P(q|x)P(c|q,x)$ [7, 8]. The context classes were estimated using a decision tree algorithm and the context class posterior was estimated using a single layer network for each phone class. A total of 604 context-dependent phone models were used. This system is described in greater detail in [9].

The acoustic models were trained solely on the prepared broadcast speech (F0) segments of the Broadcast News acoustic training data. A Viterbi training procedure was adopted.

3.2. Language Model

The system used a 65,532 word vocabulary prepared by selecting the 80,000 most frequent words from the broadcast news text data and removing misspellings, processing errors, etc. A backed-off trigram language model was built from the Broadcast News text data (132 million words), resulting in test set perplexities typically in the range 200–300.

3.3. Search

The TREC/SDR evaluation was a stress test of our recognition system, since it involved performing LVCSR over the broadcast archive (around 35 hours of speech), with some "segments" of speech up to one hour long. We have extended the NOWAY start-synchronous decoder [10], to operate in an "online" mode, decoding arbitrarily long streams of speech without an additional CPU or memory burden.

NOWAY is based on a stack decoder framework and exploits the acoustic model posterior probability estimation in an effective pruning technique referred to as phone deactivation pruning [3]. This single pass algorithm is naturally factored into time synchronous state-level processing and time asynchronous word-level processing. This enables the search to be decoupled from the language model. Incremental output of the most probable final transcription is possible owing to the tree structuring of the search and the domination of language model equivalent paths.

In this evaluation, using posterior probability based phone deactivation pruning, the usual beam pruning and a unigram language model approximation at the state level we were able to decode the evaluation broadcast archive with an average of less than 1,500 model evaluations per frame (corresponding to a run time of less than 6× real time on a Sun Ultra 1/170).

4. INFORMATION RETRIEVAL ENGINE

Version 2.0 of the PRISE system [11] was used as the information retrieval engine for this evaluation. The system was used as supplied with no modifications. The standard PRISE stop list of 23 words and the SMART stemming algorithm were used.

5. RAPID WORD SPOTTING USING POSTERIOR PROBABILITIES

CSR systems can only recognize words which are contained in their lexicon. Although the ABBOT system used for these experiments had a 65k word vocabulary, approximately 1% of the words in the test set were out of vocabulary (OOV).

This raises a potential problem at the information retrieval stage: infrequent words are potentially important during retrieval but such words are most likely to be OOV and thus could have a deleterious effect on performance. To counteract this, a rapid word spotting module was added to the system to try and find any OOV query words.

The queries were scanned for OOV words. Any OOV words for which pronunciations did not exist were sent to an automatic pronunciation generator using the letter-to-sound rules in the Festival speech synthesis system [12].

The word spotting module used the context-independent posterior probability estimates from the recurrent network acoustic model, dynamically constructing word models for target words and using a set of looped phone garbage models. Any spotted words were added into the appropriate section of the speech recognition transcription. The transcriptions were then re-indexed and the standard retrieval procedure followed[1].

In the event, the only OOV word in the test queries was 'CIA' (ABBOT treats each letter of an abbreviation as a separate word and was thus expecting C. I. A.). Furthermore, no instances of it were found by the word spotting module (because it treated it as a word rather than a string of letters). Consequently, the word spotting module had no effect on system performance during this experiment.

6. EXPERIMENTS

6.1. Speech Recognition Performance

We applied the ABBOT system to the SDR test data, consisting of around 50 hours of Broadcast News, of which around 35 hours needed to be recognized. Table 1 shows the word error rate (WER) for this data set, broken down into the seven focus conditions.

We estimate the relative search error (introduced by pruning) to be around 15%. This was very much a baseline system which made no attempt to adapt to different focus conditions, or to segment out non-speech portions from the documents (e.g., musical interludes) to reduce the number of insertions.

[1]Obviously, this technique could not be used on a large corpus or in a practical system, but it does give an indication of the importance of OOV words

Table 1: ABBOT Performance at the Broadcast News Focus Conditions

Focus	Description	WER
F0	Baseline Broadcast Speech	24.9%
F1	Spontaneous Broadcast Speech	43.2%
F2	Speech / Telephone Channels	50.8%
F3	Speech / Background Music	49.4%
F4	Speech / Degraded Acoustic Conditions	35.5%
F5	Speech / Non-Native Speakers	36.3%
FX	All other speech (combinations)	55.7%
-	Overall	40.1%

6.2. IR Performance

We compared the performance of the system using the supplied transcript, the supplied output of the baseline recognizer and the output of the ABBOT recognizer. These results are summarized in Table 2.

Table 2: TREC SDR Results for PRISE IR System

Transcription	Mean Rank	Mean Reciprocal
Reference	11.59	0.6236
Baseline Recognizer	30.43	0.5062
ABBOT LVCSR	27.82	0.5784

Due to a problem with some of the Baseline Recognizer transcriptions, two of the (87) broadcasts had to be excluded from the final analysis. Omitting these sections from the excluded broadcasts at the indexing stage (rather than removing them after the search stage) produced results that differed by less than 2% from our submitted results. Also some of the queries used a slightly different format to that expected by our system. Changing formats again resulted in a minimal change to the system performance.

We have analysed the IR performance with respect to the WER and the focus conditions. Figure 3 shows a scatter plot of retrieval rank versus WER for the baseline and ABBOT recognizers using PRISE for the 49 retrieved target sections. The plot suggests that there is a good chance of obtaining a low retrieval rank if the WER of the target section is less than about 40%.

Figure 4 graphs the mean reciprocal retrieval performance against the WER for both recognizers. Also plotted are the cumulative WER distributions for each recognizer. In this case the WER was used as a rejection threshold, and only those documents (and corresponding queries) with a WER below that threshold were considered. For the ABBOT system, about 65% of documents had a WER of 40% or less,

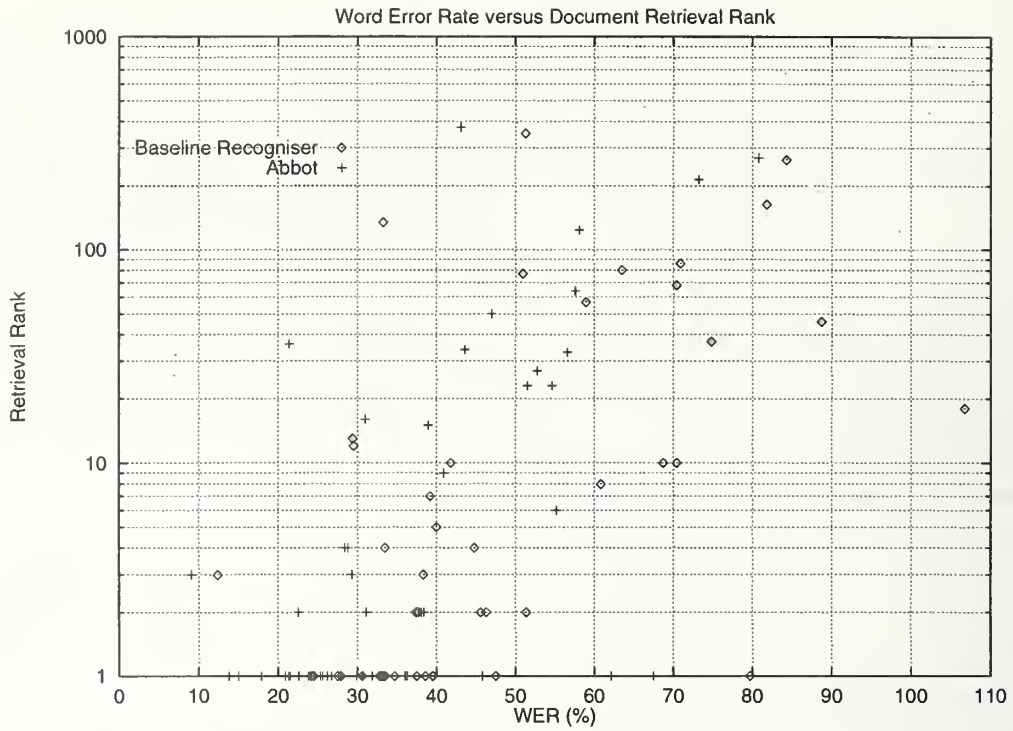


Figure 3: Document retrieval rank vs. WER.

and using those documents the mean reciprocal ranking for retrieval was around 0.75. The ROC curves reinforce the message of the scatter plot: that performance begins to fall sharply if the WER of the target document is over 40%.

Figure 5 graphs the mean reciprocal ranking against the WER for target sections containing speech largely from the F0 and FX focus conditions (twelve of each). It shows a similar picture to Figure 4: retrieval performance is good when WER is below 40%, above this figure it begins to deteriorate. Most of the F0 target sections had low WER resulting in an overall mean reciprocal figure of 0.7986 whereas some of the FX target sections had high WER contributing to an overall mean reciprocal figure of 0.6031.

7. CONCLUSION

Our principal goal in this evaluation was to develop a working spoken document retrieval system, and to apply our recognizer to tens of hours of broadcast speech data. We have succeeded in this objective. Future work will involve development of IR methodologies for spoken document retrieval (rather than treating the problem as text retrieval and using an "out-of-the-box" system) and to further improve the speech recognition component.

8. ACKNOWLEDGMENTS

Thanks to Paul Over, John Garofolo and Ellen Voorhees of NIST for help and advice with the PRISE system and the TREC evaluation. Thanks also to Alan Black of the University of Edinburgh for assistance with his Festival system.

9. REFERENCES

[1] H. Bourlard and N. Morgan, *Connectionist Speech Recognition—A Hybrid Approach*. Kluwer Academic, 1994.

[2] A. J. Robinson, "The application of recurrent nets to phone probability estimation," *IEEE Trans. Neural Networks*, vol. 5, pp. 298–305, 1994.

[3] S. Renals, "Phone deactivation pruning in large vocabulary continuous speech recognition," *IEEE Signal Processing Lett.*, vol. 3, pp. 4–6, 1996.

[4] G. Williams and S. Renals, "Confidence measures for hybrid HMM/ANN speech recognition," in *Proc. Europ. Conf. Speech Communication and Technology*, (Rhodes, Greece), pp. 1955–1958, 1997.

[5] J. Hennebert, C. Ris, H. Bourlard, S. Renals, and N. Morgan, "Estimation of global posteriors and

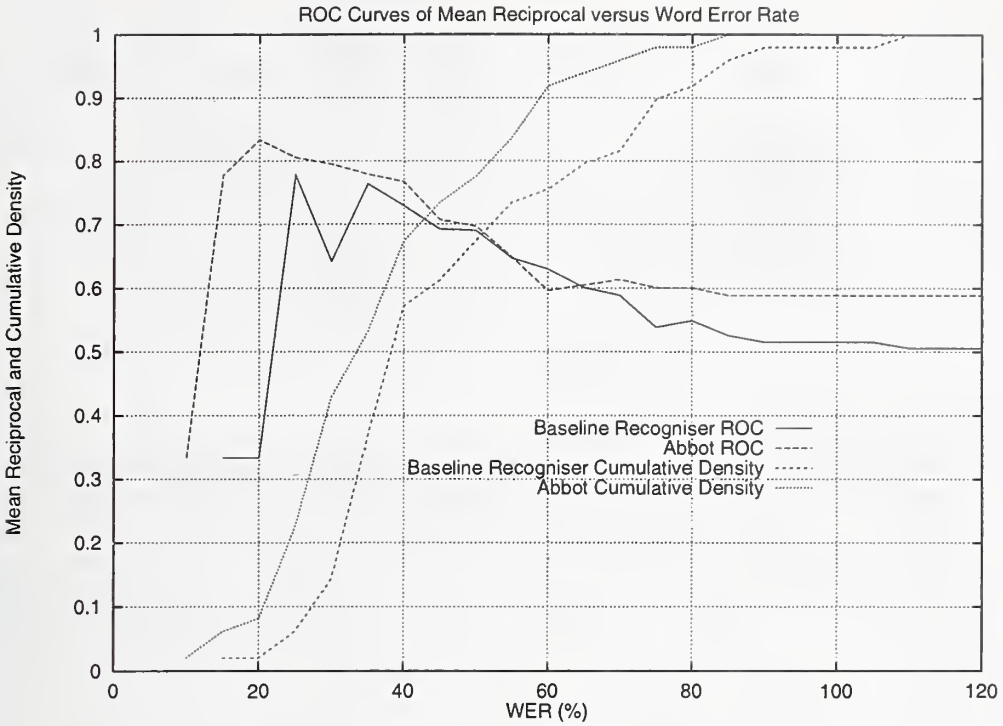


Figure 4: Mean reciprocal retrieval performance vs. WER.

forward-backward training of hybrid HMM/ANN systems," in *Proc. Europ. Conf. Speech Communication and Technology*, (Rhodes, Greece), pp. 1951–1954, 1997.

[6] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, pp. 1738–1752, 1990.

[7] H. Bourlard, N. Morgan, C. Wooters, and S. Renals, "CDNN: A context dependent neural network for continuous speech recognition," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, vol. 2, (San Francisco), pp. 349–352, 1992.

[8] D. J. Kershaw, M. M. Hochberg, and A. J. Robinson, "Context-dependent classes in a hybrid recurrent network-HMM speech recognition system," in *Advances in Neural Information Processing Systems*, vol. 8, MIT Press, 1996.

[9] G. D. Cook, D. J. Kershaw, J. D. M. Christie, C. W. Seymour, and S. R. Waterhouse, "Transcription of broadcast television and radio news: The 1996 ABBOT system," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Munich), pp. 723–726, 1997.

[10] S. Renals and M. Hochberg, "Efficient search using posterior phone probability estimates," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, vol. 1, (Detroit), pp. 596–599, 1995.

[11] D. Harman, "User-friendly systems instead of user-friendly front-ends," *Journal of the American Society for Information Science*, vol. 43, pp. 164–174, 1992.

[12] A. Black and P. Taylor, "Festival speech synthesis system: system documentation (1.1.1)," Tech. Rep. HCRC/TR-83 (http://www.cstr.ed.ac.uk/projects/festival/manual-1.1.1/festival-1.1.1.ps.gz), Human Communication Research Centre, University of Edinburgh, 1997.

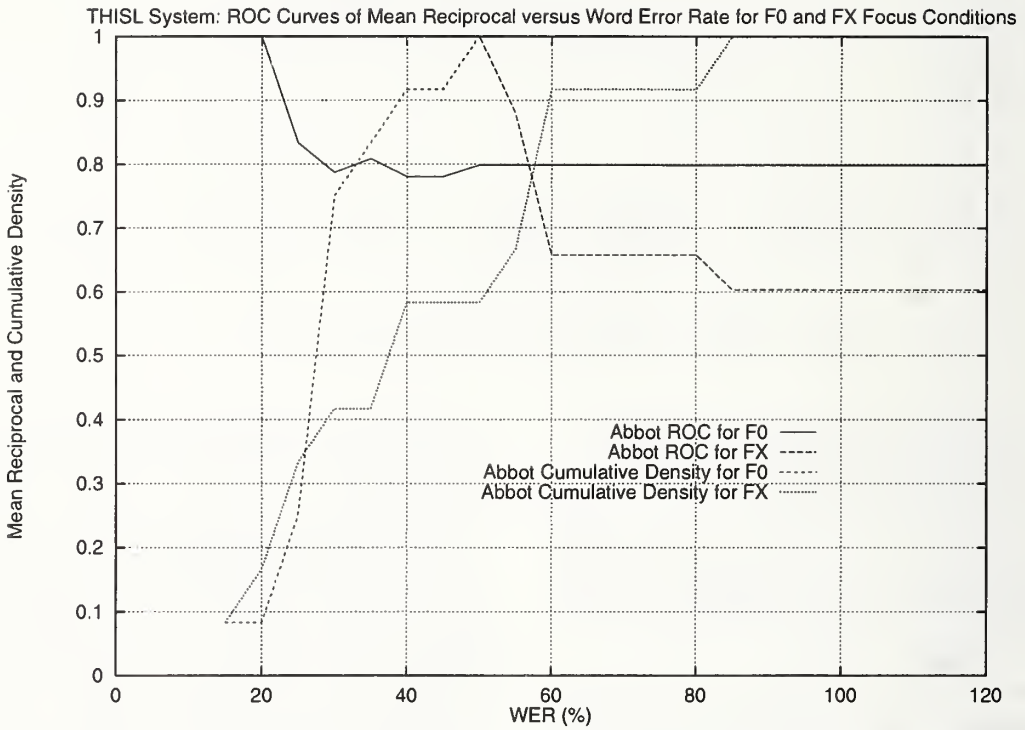


Figure 5: THISL system: mean reciprocal retrieval performance vs. WER for target documents at F0 and FX focus conditions.

Cross Language Retrieval with the Twenty-One system

Wessel Kraaij
TNO-TPD
P.O. Box 155, 2600 AD Delft
The Netherlands
kraaij@tpd.tno.nl

Djoerd Hiemstra
University of Twente/CTIT
P.O. Box 217, 7500 AE, Enschede
The Netherlands
hiemstra@cs.utwente.nl

Abstract

The EU project Twenty-One will support cross language queries in a multilingual document base. A prototype version of the Twenty-One system has been subjected to the Cross Language track tests in order to set baseline performances. The runs were based on query translation using dictionaries and corpus based disambiguation methods.

1 Introduction

1.1 Twenty-One project

Twenty-One is a 2 MECU project with 11 partners [1]funded by the EU Telematics program, sector Information Engineering. The project subtitle is "Development of a Multimedia Information Transaction and Dissemination Tool". Twenty-One started early 1996 and is currently in its building phase.

The Twenty-One database consists of documents in different languages, initially Dutch, English, French and German but extensions to other European languages are envisaged. The TREC Cross Language (CLIR) track task fits our needs to evaluate the system on the aspect of cross language retrieval performance.

1.2 TREC6

Although the development of the full scale Twenty-One system just started in the summer of 1997, Twenty-One accepted the challenge to participate in the cross language track of TREC6.

Whether we would complete the task was a complete question, because at that moment (May 1997), the TNO mono-lingual vector space search engine was still under development and untested, The delivery of a fast workstation was also delayed, and moreover, the consortium was still negotiating with two publishers to acquire bilingual dictionaries. But finally all hard-, soft- and lingware became available just in time to complete some runs in two hectic weeks, without any time for thorough testing.

1.3 Cross Language Retrieval in Twenty-One

The primary approach to Cross Language Retrieval in Twenty-One will be Document Translation (DT). There are certain advantages and disadvantages to DT:

- DT reduces the Cross Language Retrieval task to a monolingual search issue
- The quality of a translation can in principle be better because the full document context is available. In the case of query translation there is often very little context.
- Document translation is slow, but can be done off-line.
- DT requires a full translation of the document base for each supported language, which makes it not really scalable.

The DT approach in Twenty-One will be supplemented with query translation, as a fall-back option and local feedback in the target language for recall enhancement.

A more elaborate description can be found in [2]. However we will test this approach not until TREC7 because the system's partial translation module is not yet finished.

The goal of this year's TREC6 participation (our first participation) is to test the monolingual search

[1]Project partners are: Getronics software, TNO-TPD, DFKI, Rank Xerox Grenoble, University of Twente, University of Tübingen, MOOI foundation, Environ, Climate Alliance, VODO and Friends of the Earth

system and perform baseline runs with dictionary based word translation as a preparation to a full evaluation of Twenty-One within TREC7.

2 Experimental setup

2.1 Retrieval System

The Twenty-One demonstrator[2] system is based on two types of indexes:

- A fuzzy phrase index (n-gram search on phrases extracted from the documents via NLP).
- A standard Vector Space Model (VSM) index based on lemmas

The first index type is well suited for short queries and interactive query refinement, whereas the VSM index is better suited for longer queries. For TREC6 all experiments have been done with the TNO vector space engine. This index employs straightforward *tfidf* weighting and document length normalization. As preprocessing step we used the Xerox morphological tools for tokenization, Part-of-Speech (POS) disambiguation and lemmatization[3]. The dictionary part of the index used for the TREC6 experiments consists of a concatenation of lemma and POS tag. Function words were excluded from the indexing process, based on their POS tag. No traditional stopping list was used.

2.2 Bilingual dictionaries

The translation of the topics was based on a word by word translation process, using the VLIS lexical database from *van Dale* publishers. The VLIS database is a relational database which contains all lexical knowledge that is used for publishing the dictionaries Dutch → foreign language (German, French, English, Spanish). So the database is based on Dutch headwords with translation relations to equivalent lemmas in the foreign languages. The lexical material from the foreign language → Dutch companion dictionaries is not included in the VLIS database. This has some important consequences for its use in a translation system. There are three different types of language pairs:

- Translating from Dutch to a foreign language. This is essentially equivalent to taking the printed version of the van Dale dictionary and looking up each word.
- Translating from a foreign language to Dutch. Although the foreign → Dutch material is not in the database, we can simply lookup Dutch headwords that have the query term as a translation by specifying an appropriate SQL query.
- Translating between two foreign languages. This is simply a combination of the previous types. Look for words in the target language which are a translation of a Dutch lemma which in turn has the query word in the source language as its translation.

The VLIS database contains simple and composite (multi-word) lemmas for 5 languages, Dutch being the pivot language. For Dutch there are 270k entries corresponding to about 513k concepts. These concepts have translations into French, Spanish, German and English.

English	260k	40k	300k
German	224k	24k	248k
French	241k	23k	264k
Spanish	139k	28k	167k

Table 1: Number of translation relations (simple, composite and total) in the Van Dale Lexical database

For TREC6 we only used the simple lemmas. The Xerox morphological tools were used to lemmatize the words in the query in order to find translations.

2.3 Noun phrase corpus

In order to refine the crude word by word translation strategy, a list of Noun Phrases (NP) was compiled from the TREC corpus (the AP88, 89 and 90 data set). The NPs were extracted with the standard NLP tools as used in the Twenty-One system, viz. morphological analysis and POS disambiguation with the Xerox finite state tools followed by NP extraction with the TNO parser. The NPs are not just bigrams but are *maximal*, i.e. they can can contain embedded structures with conjunctions, PP-modification etc. The NPs were sorted and then counted, resulting in a list of unique phrases with frequency of occurrence. As a last step, stopwords were removed.

3 Description of runs

Because the test environment was up and running rather late, we decided to restrict tests to the En-

[2] http://twentyone.tpd.tno.nl/

[3] including compound splitting for German and Dutch

glish document base, but perform cross language experiments with the Dutch, German and French version of the topics. We used no specialized procedure to construct a query from a topic description[4], all runs were fully automatic, full topics (or their translations) were used as queries.

Here's a short description of the runs:

1. A baseline monolingual run: **tnoee**
2. A run based on the MT translated German topics, which were provided by Maryland: **tnodemt**
3. Take the preferred translation from the dictionary: **tno?e1** where ? can be 'd', 'f' or 'nl')
4. Take all translations from the dictionary, i.e. each topic word is substituted by a list of all translations from the dictionary: **tno?e2**
5. Mark the Noun Phrases in the original topic. Subsequently replace each word by a list of its translations. This results in a multitude of possible translations of each NP. The possible translations are disambiguated using the NP corpus which was described in the previous subsection. Section 4 describes the disambiguation procedure in more detail. Finally queries are constructed, either by:
   - mapping translation probabilities into term weights: **tno?e4**
   - taking the most probable translation: **tno?e3**

4 Disambiguation

Disambiguation of the translated NPs is based on candidate NPs extracted from the document base. The introduction of NPs (or any multi-word expression) in the translation process leads to two types of ambiguity: sense ambiguity and structural ambiguity (or underspecification) which are displayed in a data structure called a translation chart.

Figure 1 gives the French translation chart of the English NP *third world war*. Each word in this NP can have several translations that are displayed in the bottom cells of the chart, the so-called sense ambiguity. According to a list of French NPs there may be two candidate multi-word translations: *tiers monde* for the English NP *third world* and *guerre mondiale* for *world war*. These candidate translations are displayed in the upper cells of the chart. Because the internal structure of NPs was not available for the translation process, we can translate a full NP by decomposing it in several ways. For example *third world war* can be split up in the separate translation of either *third world* and *war* or in the separate translation of *third* and *world war*.

-		
tiers monde	guerre mondiale	
troisième tiers	monde mondiale terre	guerre bataille
third	world	war

Figure 1: translation chart of *third world war*

The chart of figure 1 represents a total of 12 possible translations of which only one is *troisième guerre mondiale*. Constructing the translation chart and finding the most probable translation was done as follows.

1. The query is tagged and NPs are extracted from it. The disambiguation procedure is only used to disambiguate the NPs from the query
2. During dictionary look-up the bottom cells of the translation chart are filled. (Later on in the project, dictionary look-up can be extended with the composite lemmas from the dictionary.)
3. The upper cells of the translation chart are filled with candidate NPs that contain words of the corresponding bottom cells. If possible translations of two (or more) cells coocurred in an extracted NP, the possible translations are treated as a candidate NP.
4. Probabilities are assigned to the candidate NPs in each cell of the translation chart. Probabilities are based on the frequency of the candidate NP in the document base and on the contents of the dictionary. In the final version of the Twenty-One system, information from parallel corpora will also be used to estimate probabilities [1].
5. Take the most probable candidate NP that contains possible translations of each word of the query NP.

[4]Query stopwords like *document* and *relevant* were not excluded

6. If there is no such candidate NP repeat step 5 for $n = 2$ candidate NPs. If there is still no match back-off to $n + 1$ NPs until a match is found.

For the example of figure 1 the algorithm has to back-off once because there is no candidate NP that covers the translation of all the words of the query NP (the top of the chart is empty). After one back-off step there is still some ambiguity left. Queries can be constructed either by mapping the probabilities of the translations into term weights or by taking the most probable translation.

5 Discussion

5.1 Results

run name	average prec.	performance relative to baseline (%)
tnoee	0.2752	100
tnode1	0.1453	53
tnode1-fix	0.1721	62
tnode2	0.0568	20
tnode2-fix	0.0977	35
tnode3	0.2090	76
tnode4	0.2013	73
tnodemt1	0.0977	35
tnofe1	0.0913	33
tnofe1-fix	0.1131	41
tnofe2	0.0477	17
tnofe2-fix	0.0498	18
tnofe3	0.1403	51
tnofe4	0.1305	47
tnonle1	0.0841	30
tnonle1-fix	0.1545	56
tnonle2	0.0733	26
tnonle2-fix	0.0972	35
tnonle3	0.1930	70
tnonle4	0.1729	62

Table 2: Results

Table 2 lists the the non interpolated average precision and the relative performance with respect to the baseline version **tnoee**.[5]

5.2 Preprocessing bugs

The results gave us reason to have another look at the translated queries for the different languages. Due to the enormous time constraints our system still contained some minor bugs that affected the CL results of all three languages, e.g. wrong handling of capital letters, hyphens, diacritial markers, etc. One of these minor bugs had major implications: the character $ (used as an escape character in one of the intermediate formats) caused a lot of not relevant hits, because it was not removed in all the runs.

In the table we included unofficial bugfix runs for the runs labelled '1' and '2'. These runs (in particular tnode2, tnofe1, tnofe2, tnonle1, tnonle2 and also the runs '5' and '6' which are not listed in the table) all suffered severely from the '$-bug'.

The lexical lookup and tokenizing process is still far from perfect though. Especially the handling of compounds, geographical names and diacritics needs to be improved for TREC7.

5.3 Fundamental problems

A first look at the translated queries also gives some indication of errors that are not due to bugs in our implementation, but due to our approach to CLIR.

multi-word expressions Not using the multi-word expressions from the van Dale lexical database is probably the most important source of errors. It leads to obvious errors like the wrong translation of e.g. *pommes de terres*. It also leads to errors in the translation of phrases that seem to exist of word by word translations, like e.g. *deuxième guerre mondiale* which is in English *second world war*. In French *mondiale* is an adjective and possible translations are *worldwide* and *global* but not the noun *world*. Of course, if the correct translation is not among the possible translations the disambiguation procedure will not find it either. (the multi-word expression *world war* does have an entry in van Dale.)

Proper names Because we did not use a module for proper name recognition, the system will try to translate them, which for instance leads to the translation of *Kurt Waldheim* into *Kurt forest home*.

Tagger errors The current system performs syntactic disambiguation before dictionary look-up (the Xerox tagger) and sense diambiguation after dictionary look-up. The Xerox tagger will make a small percentage of errors during the tagging process which leads to wrong translations. Maybe skipping syntactic disambiguation would be beneficial, because there is a final disambiguation step in the target language.

[5] The average precision has been computed on the basis of only 22 of the 25 topics

5.4 MT vs. dictionary look-up

The LOGOS MT run does underperform suprisingly. Upon closer inspection we found that a lot of its bad performance can be attributed to lack of robustness with respect to tokenization, compound handling, and most importantly by gaps in its dictionary. Common but vital topic terms like 'Parfum', 'Baumwolle' en 'Akupunktur' were left untranslated.

6 Conclusion & Outlook

We have succeeded in building a CLIR system which performs above median for most runs. We believe the performance of the monolingual system can be siginificantly improved by incorporating the latest weighting methods, tuning stoplist and some more attention to topic preprocessing.

The general picture of our CLIR runs is that taking the preferred translation from the dictionary works better than taking all translations with equal probability. But more important, the corpus based disambiguation technique seems to result in significant improvements. We don't know yet how much of this improvement is due to the phrase context. It's also not clear whether taking the most probable translation is better than taking the probability vector as the translation for each term.

Although it's easy to produce a table filled with average precision figures, it's hard to draw conclusions about the relative merits of the different systems and methods. The quality of a significant part of the topic translations provided by NIST and CLIR participants is not without errors or omissions, which makes comparisons across languages less meaningful (even comparing to the English baseline). The variance of the results among the topics is also extremely high because of gaps in the translation dictionaries. This makes a comparison of CLIR methodologies based on different dictionaries[6] an impossible task. Supplying a base-line dictionary (like the base-line Speech Recognizer results delivered by NIST in the SDR track) would enable a more meaningful comparison of dictionary based methods. Otherwise CLIR participants might find themselves comparing the coverage of their dictionaries instead of comparing methods for CLIR.

Acknowledgements

We would like to thank all colleagues working on Twenty-One . In particular we want to thank: Rudie Ekkelenkamp, Jurgen den Hartog for their work on the search engine (both TNO), Hervé Poirrier, Anne Schiller and David Hull of RXRC for help with the Xerox morphological tools and general advice, Franciska de Jong and UT students for translating the queries to Dutch, Tillman Wegst of DFKI for the integration of the Xerox morphological tools with the TNO parser.

[6] e.g. between our van Dale runs, the LOGOS MT run and runs from other groups

References

[1] Djoerd Hiemstra, Franciska de Jong, and Wessel Kraaij. A domain specific lexicon acquisition tool for cross-language information retrieval. In L. Devroye and C. Chrisment, editors, *Proceedings of RIAO'97*, pages 217–232, 1997.

[2] Wessel Kraaij. Multilingual functionality in the TwentyOne project. In David Hull and Douglas Oard, editors, *AAAI Symposium on Cross-Language Text and Speech Retrieval.* American Association for Artificial Intelligence, March 1997. http://www.clis.umd.edu/dlrg/filter/sss/papers/.

7 Questionnaire

1. OVERALL APPROACH:

1.1 What basic approach do you take to cross-language retrieval?
[X] Query Translation IN TREC6
[X] Document Translation : in the project and probably in TREC7
[ ] Other, _____

1.2 Were manual translations of the original NIST topics used as a starting point for any of your cross-language runs?
[X] No
[ ] Yes, _____

1.3 Were the automatically translated (Logos MT) documents used for any of your cross-language runs?
[X] No
[ ] Yes, _____

1.4 Were the automatically translated (Logos MT) topics used for any of your cross-language runs?
[ ] No
[X] Yes, run tnodemt1

2. MANUAL QUERY FORMULATION:

No manual query formulation.

3. USE OF MANUALLY GENERATED DATA RESOURCES:

3.1 What kind of manually generated data resources were used?
[X] Dictionaries
[ ] Thesauri
[X] Part-of-speech Lists
[X] Other, Lemmatizers

3.2 Were they generated with information retrieval in mind or were they taken from related fields?
[ ] Information Retrieval
[ ] Machine Translation
[X] Linguistic Research
[X] General Purpose Dictionaries
[ ] Other, _____

3.3 Were they specifically tuned for the data being searched (ie.

with special terminology) or general-purpose?
[ ] Tuned for data; Please specify _____
[X] General purpose

3.4 What amount of work was involved in adapting them for use in your information retrieval system.

Dictionaries: 3 days
Morphology: 3 days

3.5 Size
For dictionary size cf. table 1.in the paper.

3.6 Availability? - Please also provide sources/references!
[X] Commercial: Xerox Xelda toolkit
[X] Proprietary: Van Dale dictionaries
[ ] Free
[ ] Other, _____

4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES:

4.1 Form of the automatically constructed data resources?
[ ] Lexicon
[ ] Thesaurus
[ ] Similarity matrix
[X] Other, List of Noun Phrases extracted from the corpus

4.2 What sort of training data was used to construct them?
[X] Same data as used for searches, _____
[ ] Similar data as used for searches, _____
[ ] Other data, _____

4.3 Size
[ ] 4.4 million _____ entries
[ ] 128 MBytes

4.4 Was there any manual clean-up involved in the construction process?
[ ] Yes, _____
[X] No

4.5 Rough resource estimates for building the data resources (ie. an indicator of the computational complexity of the process).
[10] (Sparc Ultra 300 Mhz) hours
[ ] _____ MBytes of memory used
[ ] _____ temporary disk space

5. GENERAL

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?
[ ] Very dependent, _____
[ ] Somewhat dependent, _____
[X] Easily replacable, _____
[ ] Don't know

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?
[ ] Yes, a lot, _____
[X] Yes, somewhat, _____
[ ] No, not significantly, _____
[ ] Don't know

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?
[ ] Yes a lot, _____
[X] Yes, somewhat, _____
[ ] No, not significantly, _____
[ ] Don't know

5.4 Are similar resources available for other languages than those used?
[X] Yes, Spanish
[ ] No

Text Retrieval via Semantic Forests

Patrick Schone, Jeffrey L. Townsend, Thomas H. Crystal*, and Calvin Olano
U.S. Department of Defense
Speech Research Branch
Ft. George G. Meade, MD 20755-6000

I. INTRODUCTION

We approached our first participation in TREC with an interest in performing retrieval on the output of automatic speech-to-text (speech recognition) systems and a background in performing topic-labeling on such output. Our primary thrust, therefore, was to participate in the SDR track. In conformance with the rules, we also participated in the *Ad Hoc* text-retrieval task, to create a baseline for comparing our converted topic-labeling system with other approaches to IR and to assess the effect of speech-transcription errors. A second thrust was to explore rapid prototyping of an IR system, given the existing topic-labeling software.

Our IR system makes use of software called Semantic Forests which is based on an algorithm originally developed for labeling topics in text and transcribed speech (Schone & Nelson, ICASSP '96). Topic-labelling is not an IR task, so Semantic Forests was adapted for use in TREC over an eight-week period for the *Ad Hoc* task, with an additional two weeks for SDR. In what follows, we describe our system as well as experiments, timings, results, and future directions with these techniques.

II. GENERAL SYSTEM OVERVIEW

In order to do database designing and then querying, our overall system required a number of steps, as illustrated in Fig. 1. The preliminary steps, shown as the first three blocks in the figure, involve preparing the data for use with our topic-labelling software, Semantic Forests. Since Semantic Forests has not been specifically tailored for SGML applications, each database document needed first to be filtered to select out only the <TEXT> portion of each database message and then formatted for processing. Subsequently, all the words in a file which are spellings of numbers were converted to their numeric value (such as "seven" becoming "7"). For the *Ad Hoc* task, most numbers were already converted, so this stage was eliminated, although it was used for the SDR task. The second preparatory stage was to transform multiword units into single tokens (e.g., "United States" becomes "united_states"). The list of multiwords used for this process was derived both by hand, as well as from frequencies and document frequencies of commonly occurring tuples. The output from this word-joining software became the input into the main engine of Semantic Forests.

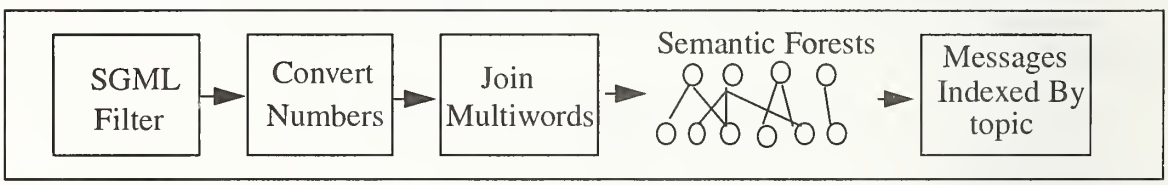
*T.H.Crystal is an integree from the IDA Center for Communications Research, Princeton, NJ.

Semantic Forests is conceptually simple. Greater detail about the algorithm will be given later. Yet suffice it for now to say that Semantic Forests reads in every word of a document, finds that word (or its stem) in an electronic dictionary, builds a topically-weighted tree based on the definition and frequencies of the word, and lastly, merges all trees together into a common graph, enhancing the scores of the words common across multiple trees. Its output is the sorted N highest scoring words from the common graph along with their weights. These N words can either come directly from text or may be generalizations or related words from the dictionary. Our goal in using Semantic Forests was to process each database message and derive its topic list, where the list itself, rather than all the words in the message, would be used to index the data.

The final stage for database-building is for an index to be built on these topic lists. Our method for doing the indexing is perhaps fairly standard. Essentially, for each unique word encountered in any topic list, there is a linked list associated. The linked list contains the message numbers, topical scores, ranks, and indicator flags for each message having that word in its topical listing. For clarity sake, it may be useful to mention that a word X has a different score and rank for each message in which it appears. To reduce the amount of seek time, each node of the linked list actually stores the information for up to four messages. This, then, completes the steps of database construction.

The methods for building both our *Ad Hoc* and SDR submissions followed this same structure, as shown in Fig. 1. Description of how we used the database to do queries will be discussed in Section V of this paper.

Fig. 1: Overall Processing Methodology



III. SEMANTIC FOREST DETAILS

Having given a general overview of our system, it may be valuable to give a more detailed view of Semantic Forests. As mentioned before, each word in a document is identified in a large recursively-closed electronic dictionary. Each of these input words may be thought of as the root node of a tree. This root node is assigned a value based upon the word's message frequency (f_{word}), a part of speech score (β_{word}), its frequency in a large corpus (F_{word}), a confidence measure of its correctness (K_{word}) and a frequency at the 50%-area mark (F_{max}) of the large corpus' *cumulative word frequency distribution*. Supposing the frequency of words were sorted from lowest to highest, the cumulative word frequency of the ith word in the sorted list is the sum of the frequencies of all words of indices less than or equal to i. If a salience score for each word is given by the near-smooth function,

$$S(word) = \begin{cases} \varepsilon + \dfrac{F_{word}}{T}\left(\log\left(\dfrac{F_{max}}{T}\right) - \varepsilon\right) & F_{word} \le T \\ \log(F_{max}/T) & T < F_{word} \le T^2 \\ \log(F_{max}/(F_{word} - T^2 + T)) & T^2 < F_{word} \le F_{max} \\ 0 & F_{max} < F_{word} \end{cases} \quad (1)$$

where ε and T are arbitrary values, representing the desired salience of a zero-frequency word and a reliability threshold frequency, then the overall score used for weighting each input word is

$$score(rootWord) = K_{word}(\beta_{word} S_{word})^{f_{word}} . \quad (2)$$

For the *Ad Hoc* task, K_{word} was set to one for every word. On the other hand, for the SDR task, the word error rate (WER) was estimated at 30%, so K_{word} was estimated by the formula $K_{word} = 1 - (0.3)^{f_{word}}$.

If an input word could not be located in the dictionary, however, rather than using ε as its salience, we pretended the word had a training frequency T. If the word was capitalized, it was assumed to be a proper noun, and otherwise it was assumed to have β_{word} of zero unless f_{word} was greater than one (in which case, β_{word} was set to be the score for nouns).

The "large corpus" of preference to be used for identifying training frequencies would have been the whole *Ad Hoc* database set. Due to time constraints, though, we were forced to use and hand-tune only the frequencies from the SDR training corpus in addition to data from some internet discussion groups. It is unknown what adverse effect this may have on system performance, but our speculation is that this caused some slight degradation.

The next step of Semantic Forests is that the words that define each input word are considered to be its children, and they receive as their score a fraction of their parent's value, based again on their individual parts of speech and large-corpus frequency, as well as on a propagation attenuation coefficient (W), their dictionary frequency (d), and the dictionary equivalent of an F_{max}, namely d_{max}. In particular, the fraction of weight that the jth child of the ith word receives is

$$fraction(i, j) = WD_j / \left(\sum_{\forall child(i) = k} D_k \right), \quad (3)$$

where D is a dictionary salience given by

$$D_j = \beta_j (S_j \log(d_{max}/d_j))^{\frac{1}{2}} . \quad (4)$$

This means that the score for the ith child word of parent word j is

$$score(\text{child word } i) = fraction(i,j) * score(\text{parent word } j). \quad (5)$$

This tree could continue to grow by augmenting the children of children and their corresponding scores, etc., but this augmentation was not done for either TREC task. (For additional details, see [1].) As can be seen, though, each word of the message is given a corresponding semantic tree structure, where the input word is the root of that tree.

The last topical step, then, is to merge all of these semantic trees into a common graph and give each word j a score $OS(j)$ as a function of its graph connections. This is illustrated in Fig. 2. As this is done, words that occur in multiple trees are strongly enhanced and are likely to be topic words or words related to the topic. Let the topical score of word j for tree i be denoted by $TS(i,j)$. Since word j may occur multiple times in a tree, we add the score for each occurrence to get $TS(i,j)$. Let $SUM(j,q)$ be the sum over i of $(TS(i,j))^q$. Then the overall score for word j is given by:

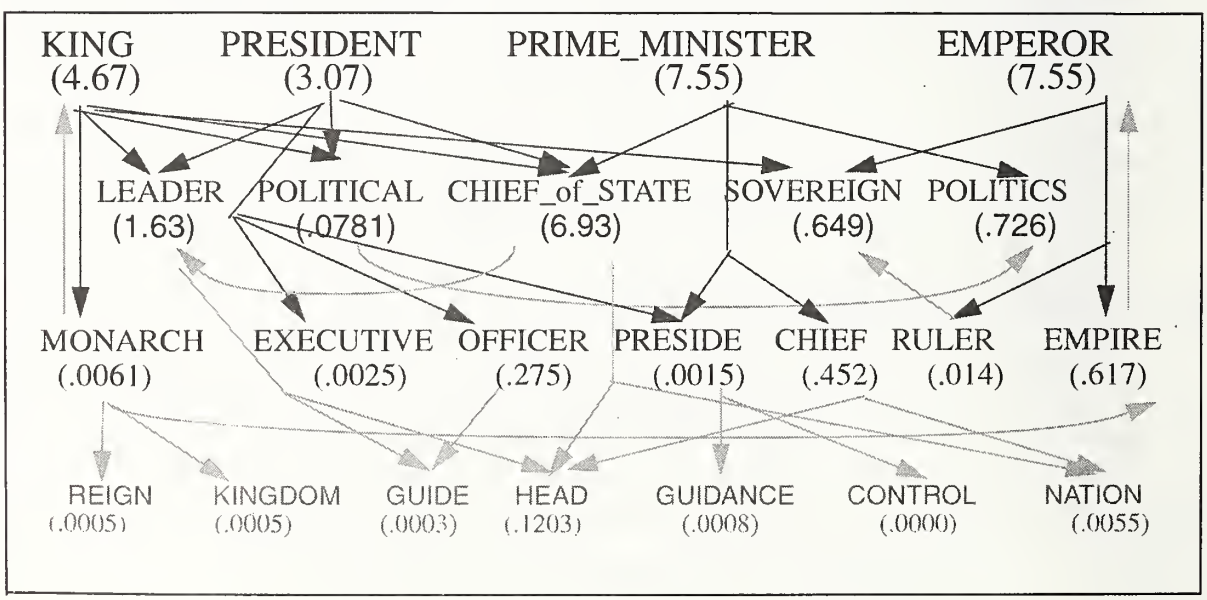
$$OS(j) = D_j^{\frac{1}{2}(\phi(j,1)+\phi(j,2))-1} \cdot SUM(j,1), \tag{6}$$

where the ϕ values come from complicated, *ad hoc* functions based on nth-order moments and maximal TS values for j. Letting $n(j)$ represent the number of trees where the TS value for j is non-zero, and letting $max(j)$ be the largest of these values, then ϕ can be given as

$$\phi(j,p) = n(j)\left(\frac{1+(-1)^p}{2} - \frac{1}{max(j)} \sum_{q=1}^{p} (-1)^q \frac{SUM(j,q)}{SUM(j,q-1)}\right), \tag{7}$$

where p is the maximum number of moments used

Fig. 2: Final weighted graph derived from merging a particular Semantic Forest.



After the common graph is created, the elements of the tree with the largest values can be selected as topics or words that are topically related. However, some pruning of the tree may be desirable.

The type of pruning we used was this: if a word of the graph is not an input word and if $n(j)$ is one, then the word is pruned from the tree. Other types of prunings are possible, but this is the only one performed for TREC purposes. After pruning, the N largest values are selected as topical representatives for the particular document, and are stored with their scores and their rank amongst the N. As is evident, this implies that words may be used in the topic descriptor even though they were not in the original document, so an indicator flag is also stored to indicate if the word was an input word or not.

A comment should be made here about implementation and memory requirements. For the sake of speed as well as memory, the results of forming a common graph were implemented as arrays of words as opposed to large sets of trees. Thus, some of the nice features about trees and graphs are lost, but most of the topical scoring ability is still retained. An example of the contents of a final array (in sorted order) is illustrated in Fig 3. For processing a single document (or query), the equivalent of eight floating point arrays the size of the dictionary (about 38K words for this task) need to be retained in addition to the dictionary itself, as well as information about any new words. This means there is a fairly low RAM overhead (about 4-7 Mbytes). On the other hand, considerable disk space is required to store the Semantic Forests output in uncompressed text form for the whole collection. For the Ad Hoc task, the required storage space was 2.38 Gbytes and for the SDR task it was 4 Mbytes. The indexing on the topical outputs from each message took much less storage. For the *Ad Hoc* task, the indices required 473 Mbytes to store; but for the SDR task, the size was only 3 Mbytes.

Fig.3 Final, sorted topic array

EMPEROR	7.547
PRIME_MINISTER	7.547
CHIEF_of_STATE	6.926
KING	4.675
PRESIDENT	3.071
LEADER	1.634
POLITICS	0.726
SOVEREIGN	0.649
EMPIRE	0.617
CHIEF	0.452
OFFICER	0.275
PRINCIPAL	0.147
ENFORCEMENT	0.142
HEAD	0.120
POLITICAL	0.078
AFFAIR	0.062

IV. TIMINGS on TRAINING

With the system structure built, the next objective was to begin processing the database messages. This was a *very* time-consuming procedure. The filters and pre-processing stages of this task were implemented in Perl scripts, making them easy to write but incredibly slow. Semantic Forests and the indexer, on the other hand, were C code optimized with a -O2 option on SunOS5. Unfortunately, since our processing on the *Ad Hoc* data was done in the week of submission, we inadvertently failed to capture the actual CPU timings for the processing. However, we do have approximate serial wall-clock timings which we believe are similar or on the high side of the actual processing time. For the pre-processing and topic-labelling stages of the *Ad Hoc* task, Table 1 on the next page gives these timings for the full 1997 *Ad Hoc* database material. The wall-clock time for indexing the topic lists was a much faster process, taking 209 minutes. Based on this fact and on the entries in Table 1, the total time for processing was approximately **85.6** hours. For the SDR task, we had two extra weeks to prepare, so we did retain actual system timings for processing. Table 2 gives the timings required for processing the smaller SDR task.

Of course, of interest is the actual platform(s) being used for processing. The two machines used for the *Ad Hoc* processing were a Sun Enterprise 5000 (250 MHz, 8 heads, and 1040 MB of RAM), which is listed in the table as machine type 1, and a SunSparc 20 (100 MHz, 2 heads, and 512 MB of RAM). Only machine type 1 was required for the smaller SDR task.

Table 1: Approximate Wall-clock Time for Topic Labelling on Ad Hoc Task

DOC TYPE	# messages	Machine Type	Pre-proc &Topic Labelling
CR-E	11,358	1	141 min.
CR-H1	7,425	1	96 min
CR-H2	9,139	1	112 min
FBIS1	61,578	1	417 min
FBIS2	26,438	1	174 min
FBIS3	42,455	2	735 min
FR1	26,843	1	201 min
FR2	28,787	1	302 min
FT1	76,857	1	683 min
FT2	133,301	1	897 min
LA1	43,803	1	361 min
LA2	54,603	1	495 min
LA3	34,210	1	311 min

Table 2: System Timing for Full Training on SDR Task

TASK	Machine Type	Pre-proc & Topic Labelling	Indexing
LTT data	1	447.89 sec	11.97 sec
SRT data	1	467.55 sec	14.20 sec

V. PERFORMING QUERIES

The next issue is to address the querying process itself as well as the scoring metrics that correspond to querying. All of the query processing was done automatically with no human intervention. For a given query, a query information extractor was first applied to the raw input which does the equivalent of what the SGML filter did before, while also using a number of regular expression searches to eliminate common non-topic phrases such as "a required document would have," as seen in the queries of previous years. Afterward, the number conversion, the multiword processing, and Semantic Forests are applied to each of the queries in the file. Thus, for each query, a topic listing is created, almost exactly as had been done for the database messages.

The goal is to determine how correlated the query topics are with each message in the database. Actual implementation of our querying scheme looks up each single token that was output of the query topic list and then finds all messages where that token was in the topic list as well. "Token," in this instance may either be a single word or a multiword unit. After each token is processed from the query topic list, though, it is as if two vectors have been correlated (namely, the vector of topics from a particular database message and the query topic list). Rather than describing what happens with each token, we will simply show how the two vectors might correlate or agree. Our score is given by

$$agreement = (hits)^{p1}\left(\sum_{\forall hit} mw^{p2} \cdot idf^{p3} \cdot F(mt, qt, mr, qr)\right). \tag{8}$$

The variable *hit* indicates a topic token agreement between the database message topic list and that of the query, and *hits*, then is just the number of these agreements over M-long topic lists (M is less than or equal to N). An agreement between a multiword unit should in general be worth more than single-word agreements, so the value *mw* is introduced. The value *mw* is set to 1 if the token in agreement is a single word, and otherwise it is the number of words in a multiword unit. Likewise, if the token in agreement is rare in terms of the number of topic lists it appears in, it should have more weight than a more frequent word. Thus, *idf* is introduced, which is simply the inverse document frequency of a word as applied to the topic lists. The function F is user-specified, as are the parameters p_i. F takes into consideration the topic ranks and the topic scores from the agreement, and is the key component of the agreement score. The variables *mt* and *qt*, represent the topic score of the particular token in the data message and in the query, respectively; likewise, *mr* and *qr* represent message and query ranks.

The values *mt* and *qt*, when output from the topic algorithm, are L_2-normalized. The rationale for this normalization was that we might want to later take a product between two topic lists and make the inner product of a list with itself be equal to one. With this notion in mind, then, the experimentation we did to develop the *F* function basically limited its structure to be

$$F(mt, qt, mr, qr) = warp((mt \cdot qt)^{p4} \otimes H(mr, qr)), \tag{9}$$

where $\otimes$ indicates some simple commutative function such as sum or multiply, "warp" indicates some non-linear function, and *H* is a type of correlation function. It seemed useful to turn rank scores into alternative types of topic scores. This can be achieved by normalizing the reverse ranks (counting backwards from *M)*; that is, if a rank is y, the normalized reverse rank would be

$$y'=(M+1-y)/BT. \tag{10}$$

The normalizing value *BT* is simply the square root of the sum of squares over the integers ranging from 1 to *M*. Thus, if we force the ranks to behave similar to the actual topic scores, the function *H* can be rewritten

$$H(mr, qr) = (mr' \cdot qr')^{p5}. \tag{11}$$

The actual final formulas and other experimentation that were performed will be explained in the next section. Table 3 shows the time it took to perform retrieval for both the *Ad Hoc* and the SDR tasks. It includes forming the topic lists for the queries and performing the score calculations. For the conditions under which we ran, wall-clock time is assumed to have been only slightly greater than CPU time.

Table 3: Query Time for Topics 301-350 and Topics SDR1-SDR50

TASK	Machine Type	Time	Timing Method
Ad Hoc: Description Only	1	11 min	wall clock
Ad Hoc: All topic info	1	26 min	wall clock
LTT: All topic info	1	8.41 sec	CPU clock
SRT: All topic info	1	8.66 sec	CPU clock

VI. EXPERIMENTATION

For our system, one of the primary 'experiments' was to get it to give reasonable answers. An initial system prototype applied to the SDR task suggested that our methodology had some poten-

tial, so construction of the full indexing-querying software was begun. We experimented with our system by using the query set from 1996 and querying against only those parts of this year's data that were also available in 1996l. All of our runs were completely automatic, so we were able to make many experiments. We compared our results against the TREC5 short and long automatic columns listed in Table 1 of Karen Sparck Jones' "Summary Performance Comparisons TREC-2, TREC-3, TREC-4, and TREC-5," noting that 25% was the lowest average precision listed for retrieval of 30 documents. As we practiced this *Ad Hoc* task, we were able to move our average precision on the top 30 documents retrieved on long automatic queries from a mere 4% at the onset to 19% at the end of experimentation. Likewise, in the SDR task, using the six sample queries, our initial protoype had an average rank of 45 and inverse of the average inverse rank of 1.5, both of which we were able to bring down substantially, getting scores as low as 4.7 and 1.37 respectively. This was done using a set of queries which included the six supplied by NIST and an additional 40 of our own queries. Thus, in this experimental phase, we learned many useful things that will also be commented upon here.

INITIAL COMMENTS

Three comments are in order at the onset. The original application of Semantic Forests, as noted previously, was to automatically label the topic of a document. This, though, is not the case for queries. When a person makes a query, he or she may not necessarily want to retrieve something whose topic matches the query. Having looked at the messages marked by NIST as relevant against some of last year's queries, we noted that some documents were so marked even if they only made passing reference to the query word or phrase. These instances would be clear losses to us since the fact of a word being in a document would not necessarily guarantee that the word would also be topical.

The second comment is that as we proceeded to do our experiments for the *Ad Hoc* experiments, we recognized that our software had two programming bugs. When these bugs were corrected, we observed a boost in our overall average precision on the top 30 (APRT30) documents by an absolute 7.5%.

Thirdly, since we were only using a subset of the total 1996 data as we performed our experiments, we expected we would be losing a few percentage points due to the fact that some of last year's queries only had documents that appeared in early databases, meaning we would automatically lose in those situations. Our expectation was that whatever our final APRT30, we could probably improve performance by 2-3% based on the fact that we would have the full 1997 data set. Also, since our output would actually be evaluated, we thought this might contribute an additional 1-2%.

EXPERIMENTS PERFORMED

All the experimentation that was performed for the *Ad Hoc* task involved either specifying the functions and parameters mentioned in section IV, or limiting or expanding the number of non-input synonym words that are generated by Semantic Forest for either the database document or the query. Our SDR experimentation also involved trying to find common errors and omissions made in the recognizer output and supplementing our electronic dictionary to take these variations into account.

Synonymy:

A common recurring theme in past TREC's is how to supplement a query with synonyms in such a way that there is a performance boost. The idea of producing conceptually similar words is something quite natural for Semantic Forests. We expected that this innate quality would be a big boon for our system. Unfortunately, Semantic Forests reports terms as found in its dictionary which, though conceptually similar, are not always synonymous. Likewise, Semantic Forest does not yet report which sense of the word is desirable. Both of these factors made it difficult to use synonyms. The general finding when using the out-of-the-box synonyms were that the messages that were already fairly well correlated suddenly had gigantic scores; but weakly correlated messages were not enhanced and were generally retrieved lower in the queue since other documents might have more synonyms. An example of both of these cases might be made from the SDR sample 6 queries. Before synonyms were added, our best query had an agreement score of 98 and a rank of 1, while our worst had a score of 0.58 and a rank of 63. After the synonyms were added, the first now had a score of 500K and the rank did not change; but the worst had a slight drop in score of 0.367 (due to the fact that synonyms can possibly rank higher than input words), and, more importantly, its rank had dropped to 127. Thus, for this TREC, we limited our synonyms to two other types.

The first was to use subcomponents of multiword units. In this instance, if a query has a multiword unit that agrees with one of the messages, there is a hit not only on the word itself but on the salient subcomponents as well. On the other hand, a database message may not have the particular multiword being sought but may have the component words. For the *Ad Hoc* query, we did not use this, though it was incorporated into the SDR experiments and resulted in dropping the inverse of the average inverse rank on our 45-query set from 1.45 to 1.37 and average rank from 5.83 to 5.58 at the particular time it was first added.

The second type of synonym has to do with adjectival nouns that reference a country. For example, "French" implies "France" and "Israeli" implies "Israel." Therefore, if an adjectival noun existed in a document and if it had in its definition a country name with the first two characters the same, the country name was added to the document and processed as if it had actually been a member of the text. For the *Ad Hoc* experiment, we used this type of synonymy on both the database documents as well as on the queries, and we experienced a positive effect. On the version of the system used to conduct the experiment, the APRT30 increased (on the full topic) from 17.5% to 19.0%. On the SDR task, this same synonymy was applied to both the query and document and to the query alone. There was a decrease in average rank when the synonymy was limited to only the query.

Parameter Modifications

For both the *Ad Hoc* and the SDR tasks, determining the best user-specified parameters and functions was difficult, but fairly useful. For the *Ad Hoc* task, the parameter modifications did results in slight improvements. We basically let $p2=0$, $p1=p3=p4=p5=1$, warp=$\sqrt{x}$, and $\otimes$ = multiply in equations (8), (9) and (11). We started with $M=250$ and tried to reduce this, but this change decreased the precision. The biggest improvements for the *Ad Hoc* came in limiting the allowable ranks. We got a 0.8% absolute improvement in APRT30 when we set qr′to zero when $qr>75$ (i.e., weeding out lesser important query topics). Also, when a hit occurred such that $mr>6$, it was counted as only 1/4 in the *hits* parameter. This gave an absolute 1.2%.

However, for the SDR task, these parameters made much larger differences. In particular and of prime interest is the fact that we eventually ended up deciding that the topic score was of no importance when compared with the rank, and it was completely eliminated. For the final system, we settled on letting $p1=3$, $p2=2$, $p3=1.5$, $p4=0$, and $p5=1$; $\otimes$ remained a simple multiply, but warp(x) was changed to $(0.85)^x$ and H, as previously defined in Equation (11), was modified to become

$$H(mr, qr) = (mr + qr)^{p5}. \qquad (12)$$

Common Recognizer Errors

As was mentioned before, we were particularly interested in the SDR track of TREC97. Our hope was that Semantic Forests could potentially knit together the true topics of errorful transcriptions, but we also hoped to be able to supplement that effort by locating commonly misrecognized words and putting into the dictionary the misrecognition. In particular, we wanted to supplement the dictionary with words that are high frequency in the LTT training files, but non-existent in the SRT. After analyzing the training data, we realized that there were not many instances of this kind of phenomenon, but the words that were missing from recognition were usually critical. In particular, "Netanyahu," "Valujet", "Freemen," and "Admiral Boorda" are very topical words that appeared in many instances across the LTT files, but never occurred in the SRT. Common misrecognitions of "Valujet," for example, were the phrases "value jet" and "valued jet;" "Netanyahu" often was recognized with the word "neon" in it, such as "neon who;" "Freeman" almost always appeared as two separate words; and "Boorda" often appeared as "border" or something similar. As an experiment, we wrote some practice queries that only involved these phrases and compared our algorithm's output to that of another retrieval system. As one might expect, our performance was far better. Unfortunately, after we had submitted our actual evaluation of SDR, we found that none of these words were in the SDR queries. On the other hand, the word "Unabomber" did appear in the queries and we were prepared.

VII. OTHER INTERESTING OBSERVATIONS

It is interesting and useful to make note of a few other observations that can be made about this approach. These areas are things that are inherent in the algorithm but may be considered experimental in their own right. These are the areas of stemming, use of numbers, and boolean logic.

Stemming

In the past, different groups have experimented with different kinds of stemming, such as using the first k characters or using some more sophisticated method. For Semantic Forests, stemming is automatic. If a word in the text document is also in the electronic dictionary, then the word is considered to already be stemmed. There are instances when this consideration is wrong, where, for example, what might be a conjugation of a verb might also have another meaning, and therefore the conjugation may also be stored in the dictionary. Yet these cases are infrequent. On the other hand, if a word is not found in the dictionary, then procedures are applied to see if it is a different word form of one of the words already in the dictionary. If the word is still not found, it is assumed to be a new word.

However, words that are considered to be new result in a great difficulty that does not exist in other stemmers. Suppose, for example, that a words "cryogenically" and "cryogenics" appear in training data but are not words that existed in Semantic Forests' dictionary (nor their stems). If a query is made about "cryogenics," only messages with that exact word will be identified. We did not realize the full extent of this problem until the SDR evaluation, where for some reason, the word "programmers" was not known to Semantic Forests. It therefore stored the whole word and missed any related or partial words, causing our recognition of that message to be abysmal.

Numbers

As was mentioned before, a number conversion routine is one of the early precursors to Semantic Forests. Semantic Forests knows what numbers are, and in fact, it can interpret years to a certain extent, even knowing some major events of those years. In the 1996 evaluation, one of the queries had asked to find some event "since 1950." Semantic Forests knows both words, but it does not understand the pairwise construct. It has not yet learned to interpret "since <date>" as "greater than or equal <date>." We did not have time to put this into Semantic Forests, so we made an attempt to fake it in the SGML filter. The phrase "since <date>" was converted to the string "<date>,..., 1996, 1997," i.e., "since 1950" became "1950, 1951,..., 1997." Semantic Forests reported that dates and numbers were the key topics. This clearly was an incorrect interpretation. As a result, for the final system, we decided to leave out any special numerical processing other than what was already in the system.

Boolean Logic

Similarly, boolean logic is something that is not yet interpreted by Semantic Forests, so the "not" logic adversely effects performance of the retrieval system. To partially remedy this, the SGML filter was told that if it saw "word1 not word2," that it should just eliminate word2 altogether. Other rudimentary facilities were added to this script which enhanced our whole query routine's ability to find documents...at least in training. Yet the other boolean constructs were basically unregarded, which could have caused negative effects on processing.

VIII. FUTURE WORK

There were a number of areas that we would like to explore but did not have the time prior to evaluation to properly pursue. Other ideas did have some initial attempts made, but though the ideas may eventually provide great benefits, these first attempts were unfruitful. In particular, then, the areas we would like to work on in the future would be:

[1] Take full advantage of the synonym property of Semantic Forests. A place where this would be of particular utility is when a query is performed and there is a particular word in the query that has no messages containing it. This would have been a sure win on the SDR task, since this potentially helps reduce the number of catastrophic failures that might arise;
[2] Apply on an *Ad Hoc* task the multiword decomposition, and use the adjectival nouns only on the queries themselves, where both of these resulted in improvements on the SDR task;
[3] Insert number parsing and boolean logic directly into Semantic Forests; and lastly
[4] Perform a second pass search which looks at the actual words of the document after the topic routine is used to reduce the search set.

IX. REFERENCES

Allan, J., Callan, J., Croft, B., Ballesteros, L., Broglio, J., Xu, J., Shu, H., "Inquery at TREC-5," Center for Intelligent Information Retrieval, Dept. of Computer Science, University of Massachusetts, Amherst, Mass.

Jones, Karen Sparck, "Summary Performance Comparisons TREC-2, TREC-3, TREC-4, TREC-5", Computer Laboratory, University of Cambridge, 10 Feb 1997.

Schone, P., Nelson, D., "A Dictionary-Based Method for Determining Topics in Text and Transcribed speech," 1996 IEEE International Conference on Acoustics, Speech, & Signals Processing, Atlanta, Georgia, May, 1996; Vol. 1, pp. 295-298.

Xerox TREC-6 Site Report: Cross Language Text Retrieval

Eric Gaussier Gregory Grefenstette David A. Hull
B. Maximilian Schulze*
Xerox Research Centre Europe†
{gaussier,grefen,hull}@xrce.xerox.com

January 9, 1998

Abstract

Xerox participated in the Cross Language Information Retrieval (CLIR) track of TREC-6. This track examines the problem of retrieving documents written in one language using queries written in another language. Our approach is to use a bilingual dictionary at query time to construct a target language version of the original query. We concentrate our experiments this year on manual query construction based on a weighted boolean model and on an automatic method for the translation of multi-word units. We also introduce a new derivational stemming algorithm whose word classes are generated automatically from a monolingual lexicon. We present our results on the 22 TREC-6 CLIR topics which have been assessed and briefly discuss the problems inherent in the cross-language IR task.

1 Introduction

Cross Language Information Retrieval (CLIR) addresses the problem of retrieving documents written in one language using queries written in another language. As document repositories grow in size and distribution, and one can consider the World Wide Web as such a repository, it is becoming more important to find solutions to this long-standing research problem [11].

Xerox participation in TREC-6 is limited to the Cross Language Information Retrieval track. We are interested in learning how well queries translated via a general purpose bilingual dictionary can perform in relation to queries written in the document language. Our experimental technique is to perform baseline monolingual retrieval using the topics provided by NIST (originally in English and manually translated into French and German) for English and French. Then, we produce automatic translations of the queries, attempting to see how closely the monolingual performance level can be approached. We explore three methods for query translation and document retrieval: manually constructed queries which use a weighted boolean model for retrieval, automatic translation of multi-word units, and

*Maximilian Schulze now works for Xerox Imaging Systems in Peabody, MA, USA and can be reached at: bschulze@xis.xerox.com

†Our research center has changed its name. We were formerly known as the Rank Xerox Research Centre.

the application of a new automatic method for derivational stemming. The purpose of this approach is to simulate the results that can be expected by a cross-language retrieval system that translates queries at run-time into the language of the pre-indexed documents stored in the system.

In the following sections, we will present the transformations that a query undergoes during our translation process, a discussion of the weighted boolean alternative to vector space retrieval, a tabular version of partial results from our CLIR runs, and a discussion of the problems that our system encountered in performing these tasks. Our experiments this year are limited to English and French. We do not work with the German topics or the German document collection.

2 Creating a Baseline for CLIR

In order to produce a baseline for comparing translated queries to original language queries, we first create monolingual versions of each target language query using the NIST-supplied translations.

The monolingual transformation of a query follows the following steps. The query text sections (Title, Description, and Narrative fields) are first isolated and then part-of-speech tagged [3, 13]. Our Xerox part-of-speech taggers[1] provide the user with lemmatized forms[10] of the words in the text. Using this tagged text we also extract entire noun phrases [12], as well as the decomposition of complex noun phrases into two word subparts. To create index terms, the individual words extracted from the query as well as the individual words[2] in noun phrases are derivationally stemmed. The stemmed versions of noun phrases are sorted in alphabetical order, as has been done in SMART since [5] in order to eliminate positional variation, and joined with an underscore to form a new index term for the query. Individual words and joined noun phrases are stored separately in newly created fields, which can be weighted in different ways.

This same treatment has been applied to documents for indexing. The addition of phrases derived in this way has improved our average precision for documents having many[3] relevant documents by 7% over baseline retrieval using simple stemmed words in past TREC experiments[9].

3 Translating queries

In this section we show the transformations performed on a sample query during translation. First, as in the monolingual case explained in the last section, the query is part-of-speech tagged, noun phrases are extracted, stopwords are removed, and words and phrases are stemmed. Then, in order to produce a translated version of the query, each of the query terms are expanded (a reversal of the stemming process) to produce all the derivational variants. Each of these variants is looked up in a general language bilingual dictionary. The

[1] http://www.xrce.xerox.com/research/mltt/Tools/pos.html.

[2] Stopwords are removed using standard IR stopword lists. See ftp://ftp.cs.cornell.edu/pub/smart

[3] More than four relevant documents. With documents with four or less relevant documents, adding phrases improves average precision by 14%.

translations are restemmed using a derivational stemmer for the target language, and the target language version of the initial query is recomposed in a TREC format.

3.1 Sample Treatment

Topic number 7 deals with *sex education.* The French title is *L'education sexuelle.* Let's imagine that this title is the entire French query and that we want to access English documents with it. The steps followed in the source language treatment of this query are the following:

- Part-of-speech tag sequence — *L'/DET éducation/NOUN sexuelle/ADJ*
- Stopword removal — *éducation sexuelle*
- Lemmatisation(inflectional lexicon) — *éducation sexuelle*
- Stemming (derivational lexicon) — *éduquer sexué*
- Noun Phrase extraction, stemming and alphabetical ordering —
  NP extracted: {*éducation/NOUN sexuel/ADJ*}
  Stemmed and ASCII ordered: *sexué_éduquer*

If this title were the entire query, then the monolingual query would consist of the following stemmed index terms: *eduquer sexue sexue_eduquer.* In order to translate this version of the query into English, the following additional steps are performed:

- For each single word, reverse stemming producing the related lemmas
  eduquer — éducation, éducatif, éducateur, éduquer
- Translation of the lemmas
  (English) *education, training, manners, educational, educative, educate, train, bring up*
- Stemming of the translated lemmas:
  educe, train, mannered, mannerism, bring_up
- Filtering of the stems obtained on the basis of their presence in the collection:
  educe, train, mannered, mannerism
- For multiword expressions, generate all possible combinations of the stems produce above
  eduquer_sexue mannerism_sex mannered_sex sex_train educe_sex mannerism_sexual mannered_sexual sexual_train educe_sexual mannerism_sexualiser mannered_sexualiser sexualiser_train educe_sexualiser

The derivational stemming algorithm used in these experiments is based on a new technique to automatically derive an approximation to derivational families using only a lexicon. Since the technique is currently under development, there are a number of problems which still need to be resolved. The algorithm is entirely automatic, meaning that like most

traditional stemming algorithms, it will make a number of stemming errors. A manual correction step is planned for the future. On the other hand, this means that the algorithm is nearly language independent (for certain language families, given a lexicon), so we can develop derivational stemmers for new languages relatively easily. This is a key advantage for cross-language text retrieval.

This automatic procedure was followed for all the 25 English and 25 French cross language topics, CL1 – CL25. The resulting queries were fed into a traditional information retrieval system implementing a vector space model for retrieval (a heavily modified version of SMART [2]). As monolingual baselines, the English versions of the NIST English topics were run over the English documents: this is our run XRCE-E2EA; and French versions of the NIST French topics were run over the French documents: XRCE-F2FA. From the NIST English topics, we generated French versions as described above and ran them over the French documents: this is run XRCE-E2FA; and likewise automatically generated English version from the NIST French topics: run XRCE-F2EA. The final A of the above runs stands for Automatic, as opposed to the manually constructed runs described in the next section.

4 Manual Runs - Weighted Boolean Model

The Xerox approach to manual query construction is based on a simplified weighted boolean model. The model assumes that each query can be divided into a number of concepts. Each concept consists of one or more terms, and the terms within a concept are combined using a weighted OR operator. The concepts are then combined using a weighted AND operator. In addition, the user is expected to assign a value of 1, 2, or 3 to each concept to indicate its importance in the query (1 = not important, 2 = important, 3 = mandatory). These values are used internally to adjust the concept weights before applying the weighted AND operator. A concept value of 3 leads to a strict boolean constraint, while lower values relax the constraint, allowing documents which do not contain any terms in the concept to be retrieved with a non-zero weight.[4] A longer description of the probabilistic weighted boolean model is provided in [7]. The detailed mathematical formulation of the operators can be found in [6], which should soon be available.

Previous experiments [7] have found that the weighted boolean model is particularly effective for cross-language text retrieval, as it addresses two important problems in query translation. A primary source of error in CLIR is translation ambiguity (a source language term can can have multiple unrelated target language translations). The boolean AND operator provides a natural form of disambiguation, since it is likely that correct translations will cooccur much more often in documents than incorrect translations. This approach has significant advantages when compared to other corpus-based and user-based disambiguation strategies. The search corpus itself is used for disambiguation, so domain relevance is guaranteed and no additional reference corpora are required. User knowledge is incorporated implicitly in the query construction process, so no additional user effort specifically for disambiguation is required. The boolean model also performs automatic normalization of

[4] In cases where the retrieved set was less than 1000 documents, strict boolean constraints were systematically relaxed until at least 1000 documents were returned. This step was taken solely to optimize performance for TREC evaluation, since there is no advantage to returning fewer than 1000 documents.

term importance. For example, a common source language term may have one or more rare target language translations which receive large term weights. The boolean operators make sure that rare terms and terms with many synonymous translations do not dominate during retrieval (since they may well be incorrect translations). These problems can also be addressed with other disambiguation strategies, but other methods tend to add substantial overhead to the query translation process.

For the TREC-6 CLIR experiments, two manual query sets were built, one in English, one in French. Each query set was constructed independently by a native speaker of the language who took about 2 hours to build the entire query set (slightly less than 5 minutes per query). The queries were used for both monolingual and cross-language retrieval (English to French or French to English). The English searcher is an expert with the weighted boolean system while the French searcher was using the system for the first time, although he has several years of experience as a researcher in information retrieval. Both searchers are authors of this paper and the queries were not generated in a controlled experimental setting. The manual queries average 10-12 words per topic, as opposed to 24-26 words per topic for the automatic versions. We should note that no documents were examined in the course of writing these queries, so there is no use of relevance feedback or similar techniques for term selection.

English: Is wine consumption/production rising or decreasing world-wide?
English Boolean:

3	wine wines
2	consume consumption produce production
2	increase decrease rate curve forecast future
1	world international

French Boolean:

3	vin
3	consommation production consommation_vin production_vin
2	augmentation diminution diminuer croissance croître decroître
2	monde pays

Table 1: Natural language and structured versions of query CL19.

The manually generated queries for topic CL19 are presented in Table 1. For this topic, both searchers used more or less the same concept structure for their queries, although this is often not the case. The concept weights are different, however, with the French searcher requiring that either production and consumption occur in the document for it to be retrieved.

5 TREC-6 Results for 22 Topics

Xerox submitted 8 runs to the CLIR track, consisting of all combinations of the factors: monolingual and cross-language, English and French, and Manual and Automatic. We did not submit any German runs due to time pressure but hope to work with the German

	Run	Man(WTB)	Auto	Run	Man(WTB)	Auto
original	E2E(ML)	0.437/0.595	0.411/0.572	F2F(ML)	0.407/0.562	0.394/0.540
revised	F2E(CL)	0.239/0.350	0.222/0.358	E2F(CL)	0.242/0.337	0.167/0.290
original	F2E(CL)	0.218/0.332	0.185/0.285	E2F(CL)	0.195/0.278	0.163/0.265

A2B - A = query language, B = document language (E = English, F = French)
ML = monolingual, CL = cross-language
Man(WTB) = manual (weighted boolean), Auto = automatic
#/# - Avg uninterpolated precision / Avg precision at 5, 10, 15, 20 docs (AP20)

Table 2: Average/high precision score table averaged over 22 evaluated topics.

data next year. The results presented here are averages of the 22 topics[5] which had been evaluated at the time the final version of this paper was being written. At the time of the TREC conference, only 13 topics had been evaluated. To see evaluation scores for the 13-topic set, please consult our draft paper in the notebook distributed at the conference. The average performance figures for these topics for both average uninterpolated precision and precision averaged at 5, 10, 15, and 20 documents retrieved (AP20) are presented in Table 2. In general, we feel that the high-precision measure is more appropriate for the cross-language retrieval task, due to the added burden of translating/glossing/reading documents in a foreign language.

In general, our monolingual performance is very good. While we don't have the results for the 22-topic set for all participants, our system scored at or near the top in monolingual performance for the 13-topic set in English and French. We are using the basic Okapi TREC-3 term-weighting formula and no additional query expansion other than what is provided by the derivational morphology. This seems to indicate that the derivational stemming is working well for monolingual retrieval. We have heard from other groups that traditional methods for query expansion are not helpful (or are even harmful) on this query set. We feel that derivational expansion is likely to be more robust than traditional query expansion techniques (it is actually just the inverse of stemming), so this may be part of the reason for our success. We have not yet run internal comparisons of our TREC runs with our standard stemming algorithm based on inflectional morphology.

Readers will note that Table 2 has two different lines for cross-language retrieval, original and revised. The original line represents our TREC-6 submission. The revised line gives the performance after correcting a bug in our query translation algorithm. This bug caused some terms in the translated queries to be stemmed in a different way than the documents. Any term which suffered from this problem was useless for retrieval, since it didn't match any terms in the documents. Fortunately, the number of terms affected was small, but correcting the bug does result in a noticeable increase in our cross-language evaluation scores across the board.

Our revised cross-language scores are roughly 60% of monolingual performance. This ratio is nearly identical to what we have found in our previous studies [8]. Therefore, our

[5] Actually the results are based on only 21 topics, since topic 8 has no relevant topics in English and topic 22 has no relevant documents in French. Each of these topics is excluded when averaging scores over the respective language.

new translation strategy for phrases doesn't seem to be giving us any dramatic improvement in cross-language performance. When we have the chance to test the performance of our system without this feature, we'll have a more direct measure of its impact. This ratio is low compared to the numbers reported by many other TREC-6 CLIR participants at the conference. However, many of these systems started with a much lower monolingual baseline, so in absolute terms the performance of our cross-language system is reasonable. We recognize that there is still a lot of room for improvement. Note that we perform no disambiguation of our dictionary entries. For the weighted boolean model (manual runs), we believe that there is no advantage to disambiguation, but we are currently exploring several corpus-based disambiguation strategies to be used with our vector models (automatic runs).

The manually constructed topics perform slightly better than the automatic topics in all cases, but there is no evidence that the weighted boolean model (used with the manual runs) is more effective for cross-language retrieval. Therefore, we find no experimental confirmation of the advantages we hypothesized for the weighted boolean model. Only the difference between monolingual retrieval and cross-language retrieval is statistically significant. In our previous study on English-Spanish cross-language retrieval using the Elnorte collection [7], we found a similar difference in monolingual performance but a much larger difference in cross-language performance. However, it is important to highlight the difference in experimental set-up between the two experiments. The English-Spanish dictionary was cleaned up specifically for the experiments, which was not the case this year. This is reflected in a much smaller difference between monolingual and cross-language performance in the Spanish experiments.

Note that the structure of our topics makes comparisons between the manual monolingual and cross-language runs difficult. This is because the monolingual results and the cross-language results for a single document collection are based on queries written by different searchers, which means that variation due to query formulation is confounded with variation due to query translation. Similarly, monolingual and cross-language runs using the same manual queries are applied to completely different document collections, which means that variation due to the type and density of relevant documents and the document language is confounded with variation due to query translation. In addition, comparisons between the weighted boolean model and the vector space model are more tenuous because of the differences in query formulation strategies. In our previous experiments, we used exactly the same query terms (without boolean structure) for measuring performance of the vector space model.

	Fr $\rightarrow$ En	En $\rightarrow$ Fr
ML-CL ≤ 0.10	5	5
$0.10 <$ ML-CL < 0.20	10	4
ML-CL ≥ 0.20	6	10
CL $= 0.0$	1	1

Table 3: Topic comparison of automatic monolingual (ML) and cross-language (CL) retrieval as a function of absolute difference in AP20.

With the small topic sample, it may make more sense to look at performance differences on individual topics. Table 3 splits up topics according to the absolute difference in AP20

between automatic monolingual and cross-language runs. Note that the manual runs are not included for the reasons mentioned in the previous paragraph. There are only 19 topics in the English-French column because the two topics with an average precision of 0.0 for monolingual French retrieval are deleted. From this table, we note that the performance of our system seems to be worse from English to French than from French to English. This corresponds with the fact that our English to French dictionary is less complete. We are relieved to note that there is only one query in each direction where performance is disastrously poor (no relevant documents in the top 20 when relevant documents are found in monolingual retrieval).

6 Problems Inherent in CLIR

A rapid evaluation of performance over the 22 topics evaluated by NIST reveals a number of weaknesses in the techniques or resources that we employed in the CLIR track.

6.1 Dictionary Coverage

Using a bilingual dictionary as the only source of translation alternatives puts the burden of discovering at least one correct translation equivalent on that resource. The same can be said, of course, for using parallel texts. When a word is not found in the dictionary, some default strategy should be used. Our strategy is to pass the original source word through as the target translation. In the case of proper names written in the same way in both languages, this strategy should be successful. Another (more robust) strategy implemented by Davis [4] tries to discover potential cognates in the target index using an edit distance up to two characters. This technique would capture both slight alterations of proper name spellings, as well as some true cognates if the languages are etymologically close.

We used a version of the *Oxford University Press/Hachette English-French, French-English* for the submitted TREC runs. Working directly from the SGML-markupped version of the dictionary, we derived an electronic version, limited to single word expressions. The SGML markup, in the version we had, was not always consistent, leading to a large quantity of noise in our electronic dictionary, especially in common words whose entries were the most complex, as well as patches of silence. For example, we discovered that our electronic version was missing an entry for the word *potato*, the main subject of topic CL17. We did not correct this error for our submitted TREC runs, and our experiments should be considered what one can expect when using a noisy dictionary[6].

6.2 Translation of Non-compositional Phrases

As used in these experiments, our dictionary performs only single word translation, modulo derivational variants. Before accessing the dictionary, we derivationally stem the word to be looked up, then generate all other lemmatised form of the same word, and concatenate the dictionary entries for all of these words. An equivalent strategy would be to derivationally

[6]We have begun to replace this dictionary with a cleaner though less complete multilingual dictionary available through the European Language Resources Association. http://www.icp.grenet.fr/ELRA.

normalize all the head words in the dictionary and conflate the translations of all the words stemming to the same form.

When we use this technique to translate phrases, such a technique only works if the phrases themselves are compositional phrases across the languages. For example, *éducation sexuelle* (CL7) can be translated word-by-word, modulo derivational variation, to *sex education.* But *ours en peluche* (CL25) cannot be translated compositionally into *teddy bear*, since *peluche* only translates to *plush* and *fluffy.* One solution to this problem is to have an exhaustive list of the non-compositional phrases of a language, with their translations, and a mechnanism [1] for recognizing instances of these expressions. We plan to incorporate this in future experiments for non-compositional expressions that are contained in our translation dictionaries.

In addition to derivational variants, it would be useful to include close synonyms to palliate word choice variations. For example, the French version of CL6 dealing with *air pollution* uses the rare term *pollution de l'atmosphère* whose derivational variants appear in the corpus one-half as frequently as the more common *pollution de l'air.* Similarly, *organic farming* (CL12) was described in the French version as *agriculture écologique* rather than the more common, compositionally translatable *agriculture biologique* (which appears 75 times in the French newsire versus just 6 occurrences for *agriculture écologique*).[7]

6.3 Stemming

In a general language dictionary, headwords often correspond to only one member of a derivational family. To recover other related words, we have implemented a derivational stemmer for the languages that we have been treating. Since this derivational stemmer is in its first iteration, we have seen a number of words that are over-stemmed, introducing noise into our results. The query translation program is complex, first generating derivational variants, then translating the variants, and finally stemming the resulting translations. We have discovered a bug in the last stemming step, leading to some differences between how words are stemmed in the collection and how they are stemmed in the translated queries. This results in a substantial over-generation of target language terms, since we make no attempt to disambiguate during the translation process. In most cases, this is not harmful, but we need to explore more carefully whether disambiguation might be helpful. We used the first experimental version of the derivational stemmer, and it can certainly benefit from further work.

7 Conclusion

Our conclusions at this point can only be partial. We still need to perform more comprehensive tests to measure the impact of each different component in our system, and the small topic set will limit our ability to measure significant differences. Though cross language retrieval is feasible in many cases using the existing technology, many steps in the treatment that are specific to cross language information retrieval remain to be mastered: proper level of derivational stemming, production of translation alternatives, translation of

[7] Since one of the translations of *biologique* is organic and one of the translations of *agriculture* is farming.

unknown words, translations of non-compositional phrases, proper handling of the weighting of retained translation alternatives. The work done so far only highlights a few areas where possibilities of improvement is evident, and more experimentation has to be performed to clarify the respective importance of each translation step.

References

[1] Daniel Bauer, Frederique Segond, and Annie Zaenen. Locolex: The translation rolls off your tongue. In *Proceedings of the ACH/ALLC '95*, Santa Barbara, California, July 11–15 1995.

[2] Chris Buckley. Implementation of the smart information retrieval system. Technical Report 85-686, Cornell University, 1985. SMART is available for research use via anonymous FTP to ftp.cs.cornell.edu in the directory /pub/smart.

[3] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, April 1992.

[4] Mark Davis. New experiments in cross-language text retrieval at NMSU's computing research lab. In *The 5th Text Retrieval Conference (TREC-5)*, 1997. To appear.

[5] Joel L. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods.* PhD thesis, Cornell University, September 1987.

[6] David A. Hull. A probabilistic model for the approximate matching of boolean constraints. Being cleared for publication, 1997.

[7] David A. Hull. Using structured queries for disambiguation in cross-language information retrieval. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, 1997. To appear.

[8] David A. Hull and Gregory Grefenstette. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proc. of the 19th ACM/SIGIR Conference*, pages 49–57, 1996.

[9] David A. Hull, Gregory Grefenstette, Eric Gaussier, B. Maximilian Schulze, Hinrich Schütze, and Jan Pedersen. Xerox TREC-5 site report:routing, filtering, nlp, and spanish tracks. In D.K. Harman, editor, *The Fifth Text REtrieval Conference (TREC-5)*. U.S. Department of Commerce, 1997. NIST Special Publication 500.

[10] Lauri Karttunen, Ronald M. Kaplan, and Annie Zaenen. Two-level morphology with composition. In *Proceedings COLING '92*, pages 141–148, Nantes, France, August 23–28 1992.

[11] Gerard Salton. Automatic processing of foreign language documents. *Journal of the American Society for Information Science*, 21:187–194, 1970.

[12] Anne Schiller. Multilingual finite-state noun phrase extraction. In *Workshop on Extended finite state models of language*, Budapest, Hungary, Aug 11–12 1996. ECAI'96.

[13] Anne Schiller. Multilingual part-of-speech tagging and noun phrase mark-up. In *15th European Conference on Grammar and Lexicon of Romance Languages*, University of Munich, Sept 19–21 1996.

A CLIR TRACK QUESTIONNAIRE: XEROX

CLIR TRACK QUESTIONNAIRE:

---

1. OVERALL APPROACH:

1.1 What basic approach do you take to cross-language retrieval?
[X] Query Translation
[ ] Document Translation
[ ] Other, _____

1.2 Were manual translations of the original NIST topics used as a starting point for any of your cross-language runs?
[ ] No
[X] Yes, XRCECLF2EA XRCECLE2FA

1.3 Were the automatically translated (Logos MT) documents used for any of your cross-language runs?
[X] No
[ ] Yes, _____

1.4 Were the automatically translated (Logos MT) topics used for any of your cross-language runs?
[X] No
[ ] Yes, _____

2. MANUAL QUERY FORMULATION:

2.1 If query formulation involved manual effort, how fluent was the user in the source (query) language?
[X] native speakers,
XRCECLF2EM XRCECLE2EM XRCECLF2EM XRCECLF2FM

2.2 If query formulation involved manual effort, how fluent was the user in the target (document) language?
[ ] does not apply

3. USE OF MANUALLY GENERATED DATA RESOURCES:

3.1 What kind of manually generated data resources were used?
[X] Dictionaries
[ ] Thesauri
[X] Part-of-speech Lists
[X] Other, part-of-speech taggers, lemmatizers, noun phrases extractors

3.2 Were they generated with information retrieval in mind or were they taken from related fields?
[ ] Information Retrieval
[ ] Machine Translation
[X] Linguistic Research
[X] General Purpose Dictionaries
[ ] Other, _____

3.3 Were they specifically tuned for the data being searched (ie. with special terminology) or general-purpose?
[ ] Tuned for data; Please specify _____
[X] General purpose

3.4 What amount of work was involved in adapting them for use in your information retrieval system.
[X] None
[ ] _____

3.5 Size
[X] 38,000 entries
[X] 1 MBytes compressed

3.6 Availability? - Please also provide sources/references!
[ ] Commercial
[X] Proprietary, Oxford University Press
[ ] Free
[ ] Other, _____

4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES: None

5. GENERAL

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?
[ ] Very dependent, _____
[ ] Somewhat dependent, _____
[X] Easily replacable, _____
[ ] Don't know

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?
[X] Yes, a lot, _____
[ ] Yes, somewhat, _____
[ ] No, not significantly, _____
[ ] Don't know

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?
[ ] Yes a lot, _____
[X] Yes, somewhat, _____
[ ] No, not significantly, _____
[ ] Don't know

5.4 Are similar resources available for other languages than those used?
[X] Yes, ELRA dictionaries for English-French-German-Spanish-Italian
[ ] No

APPENDIX A

This appendix contains the evaluation results for the TREC-6 runs. The initial pages list each of the runs (identified by the run tags) that were included in the different tasks/tracks. Associated with each tag is the organization that produced the run and additional information such as whether the queries were produced manually or automatically as appropriate. Following the run list is a description of the evaluation measures used for the main tasks and many of the tracks. When a track uses different measures, the evaluation measures are described in the track report. The remainder of the appendix contains the evaluation results themselves, in the order given in the run list.

ADHOC RUNS

CATEGORY A DATA

Tag	Organization	Query Method	Topic Length
aiatB1	Apple Research Labs, Apple Computer	automatic	title
att97as	AT&T Labs Research	automatic	title
city6at	City University	automatic	title
csiro97a3	Commonwealth Scientific & Industrial Research Organization	automatic	title
DCU97vs	Dublin City University	automatic	title
Mercure3	Institut de Recherche en Informatique de Toulouse (IRIT)	automatic	title
iss97vs	The Institute of Systems Science	automatic	title
LNaVryShort	Lexis-Nexis	automatic	title
mds603	MDS, RMIT	automatic	title
pirc7At	Queens College, CUNY	automatic	title
glair61	University of Glasgow	automatic	title
uwmt6a1	University of Waterloo	automatic	title
aiatA1	Apple Research Labs, Apple Computer	automatic	short
att97ac	AT&T Labs Research	automatic	short
att97ae	AT&T Labs Research	automatic	short
anu6ash1	Australian National University	automatic	short
city6ad	City University	automatic	short
csiro97a2	Commonwealth Scientific & Industrial Research Organization	automatic	short
Cor6A1cls	Cornell University	automatic	short
Cor6A2qtcs	Cornell University	automatic	short
DCU97snt	Dublin City University	automatic	short
gerua3	GE/Rutgers/Lockheed Martin/SICS	automatic	short
gmu97au1	George Mason University	automatic	short
gmu97au2	George Mason University	automatic	short
ibmg97a	IBM T.J. Watson Research Center (Brown)	automatic	short
ibms97a	IBM T. J. Watson Research Center (Roukos)	automatic	short
Mercure2	Institut de Recherche en Informatique de Toulouse (IRIT)	automatic	short
iss97s	The Institute of Systems Science	automatic	short
LNaShort	Lexis-Nexis	automatic	short
mds601	MDS, RMIT	automatic	short
jalbse0	MIT/IBM Almaden Research Center	automatic	short
jalbse	MIT/IBM Almaden Research Center	automatic	short
nsasg2	NSA Speech Technology Branch	automatic	short
pirc7Ad	Queens College, CUNY	automatic	short
Brkly21	University of California, Berkeley	automatic	short
glair64	University of Glasgow	automatic	short
umcpa197	University of Maryland, College Park	automatic	short
INQ401	University of Massachusetts, Amherst	automatic	short
ispa2	University of North Carolina	automatic	short
uwmt6a2	University of Waterloo	automatic	short
VrtyAH6a	Verity, Inc.	automatic	short

ADHOC RUNS (Continued)

CATEGORY A DATA (Continued)

Tag	Organization	Query Method	Topic Length
anu6alo1	Australian National University	automatic	long
city6al	City University	automatic	long
csiro97a1	Commonwealth Scientific & Industrial Research Organization	automatic	long
Cor6A3cll	Cornell University	automatic	long
DCU97lnt	Dublin City University	automatic	long
DCU97lt	Dublin City University	automatic	long
ibmg97b	IBM T.J. Watson Research Center (Brown)	automatic	long
Mercure1	Institut de Recherche en Informatique de Toulouse (IRIT)	automatic	long
mds602	MDS, RMIT	automatic	long
nmsu1	New Mexico State University	automatic	long
nsasg1	NSA Speech Technology Branch	automatic	long
pirc7Aa	Queens College, CUNY	automatic	long
Brkly22	University of California, Berkeley	automatic	long
INQ402	University of Massachusetts, Amherst	automatic	long
ispa1	University of North Carolina	automatic	long
VrtyAH6b	Verity, Inc.	automatic	long
anu6min1	Australian National University	manual	
CLREL	CLARITECH Corporation	manual	
CLAUG	CLARITECH Corporation	manual	
fsclt6	FS Consulting, Inc.	manual	
fsclt6t	FS Consulting, Inc.	manual	
fsclt6r	FS Consulting, Inc.	manual	
gerua1	GE/Rutgers/Lockheed Martin/SICS	manual	
gerua2	GE/Rutgers/Lockheed Martin/SICS	manual	
gmu97ma1	George Mason University	manual	
gmu97ma2	George Mason University	manual	
harris1	Harris Information Systems Division	manual	
iss97man	The Institute of Systems Science	manual	
LNmShort	Lexis-Nexis	manual	
nmsu2	New Mexico State University	manual	
Brkly23	University of California, Berkeley	manual	
glair62	University of Glasgow	manual	
uwmt6a0	University of Waterloo	manual	

CATEGORY B DATA

Tag	Organization	Query Method	Topic Length
jhuapln	Johns Hopkins University/APL	automatic	short
jhuapls	Johns Hopkins University/APL	automatic	short
unc6aas	University of North Carolina	automatic	short
unc6aal	University of North Carolina	automatic	long
unc6ma	University of North Carolina	manual	

ROUTING RUNS

CATEGORY A DATA

Tag	Organization	Query Method
att97rc	AT&T Labs Research	automatic
att97re	AT&T Labs Research	automatic
cir6rou1	Center for Information Research, Russia	automatic
city6r1	City University	automatic
city6r2	City University	automatic
CLCOMB	CLARITECH Corporation	automatic
CLMAX	CLARITECH Corporation	automatic
Cor6R1cc	Cornell University	automatic
Cor6R2qtc	Cornell University	automatic
csiro97r1	Commonwealth Scientific & Industrial Research Organization	automatic
csiro97r2	Commonwealth Scientific & Industrial Research Organization	automatic
dbulm1	Daimler Benz Research Center Ulm	automatic
gerou1	GE/Rutgers/Lockheed Martin/SICS	automatic
gesri2	GE/Rutgers/Lockheed Martin/SICS	automatic
Mercure4	Institut de Recherche en Informatique de Toulouse (IRIT)	automatic
virtue3	NEC Corporation	automatic
pirc7R1	Queens College, CUNY	automatic
pirc7R2	Queens College, CUNY	automatic
rutLADc1	Rutgers University	automatic
ETH6R1	Swiss Federal Institute of Technology (ETH)	automatic
ETH6R2	Swiss Federal Institute of Technology (ETH)	automatic
Brkly19	University of California, Berkeley	automatic
Brkly20	University of California, Berkeley	automatic
UCSDrt6	University of California, San Diego	automatic
INQ404	University of Massachusetts, Amherst	automatic
ispr1	University of North Carolina	automatic
ispr2	University of North Carolina	automatic
VrtyRT6	Verity, Inc.	automatic
rutLADw1	Rutgers University	manual
srige1	SRI International	manual
INQ403	University of Massachusetts, Amherst	manual
uwmt6r0	University of Waterloo	manual
uwmt6r1	University of Waterloo	manual

CATEGORY B DATA

Tag	Organization	Query Method
teklis	Siemens AG	automatic

TRACKS

CHINESE

Tag	Organization	Query Method	Topic Length
pirc7Ct	Queens College, CUNY	automatic	title
pirc7Cd	Queens College, CUNY	automatic	short
INQ4ch1	University of Massachusetts, Amherst	automatic	short
INQ4ch2	University of Massachusetts, Amherst	automatic	short
itich3	Information Technology Institute, Singapore	automatic	short
iss97CmD	The Institute of Systems Science	automatic	short
iss97CbD	The Institute of Systems Science	automatic	short
iss97CsD	The Institute of Systems Science	automatic	short
CLARITcAS	CLARITECH Corporation	automatic	short
pirc7Ca	Queens College, CUNY	automatic	long
ETHccA	Swiss Federal Institute of Technology (ETH)	automatic	long
itich1	Information Technology Institute, Singapore	automatic	long
itich2	Information Technology Institute, Singapore	automatic	long
city97c1	City University	automatic	long
city97c2	City University	automatic	long
city97c3	City University	automatic	long
BrklyCH3	University of California, Berkeley	automatic	long
mds607	MDS, RMIT	automatic	long
mds608	MDS, RMIT	automatic	long
mds609	MDS, RMIT	automatic	long
CLARITcAL	CLARITECH Corporation	automatic	long
Cor6CH1sc	Cornell University	automatic	long
Cor6CH2ns	Cornell University	automatic	long
UdeMbi	University of Montreal	automatic	long
UdeMseg	University of Montreal	automatic	long
uwmt6c0	University of Waterloo	manual	
ETHccM	Swiss Federal Institute of Technology (ETH)	manual	
BrklyCH4	University of California, Berkeley	manual	
CLARITcM	CLARITECH Corporation	manual	

CROSS-LANGUAGE RUNS

MONO-LINGUAL ENGLISH RUNS

	Tag	Organization	Topic Language	Document Language
*	Cor6EEsc	Cornell University	English	English
	97lsiSEE	Duke/U.Colorado/Bellcore	English	English
*	97lsiLEE	Duke/U.Colorado/Bellcore	English	English
*	ETHee1	Swiss Federal Institute of Technology (ETH)	English	English
*	TNOee	TwentyOne Consortium	English	English
	XRCECLE2EA	Xerox Research Centre Europe	English	English
*	XRCECLE2EM	Xerox Research Centre Europe	English	English

MONO-LINGUAL FRENCH RUNS

	Tag	Organization	Topic Language	Document Language
*	CEAff	CEA/DIST/SMTI	French	French
*	Cor6FFsc	Cornell University	French	French
	DCU97Fv1	Dublin City University	French	French
*	DCU97Fv2	Dublin City University	French	French
	97lsiSFF	Duke/U.Colorado/Bellcore	French	French
*	97lsiLFF	Duke/U.Colorado/Bellcore	French	French
	MercureFFs	Institut de Recherche en Informatique de Toulouse	French	French
*	MercureFFl	Institut de Recherche en Informatique de Toulouse	French	French
*	clcrl1	New Mexico State University	French	French
*	ETHff1	Swiss Federal Institute of Technology (ETH)	French	French
*	CLIPS1	University of Montreal	French	French
	CLIPS2	University of Montreal	French	French
	CLIPS3	University of Montreal	French	French
*	XRCECLF2FM	Xerox Research Centre Europe	French	French
	XRCECLF2FA	Xerox Research Centre Europe	French	French

MONO-LINGUAL GERMAN RUNS

	Tag	Organization	Topic Language	Document Language
*	97lsiLGG	Duke/U.Colorado/Bellcore	German	German
	97lsiSGG	Duke/U.Colorado/Bellcore	German	German
*	ETHdd1	Swiss Federal Institute of Technology (ETH)	German	German
*	BrklyG2GA	University of California, Berkeley	German	German
	umcpxgg1	University of Maryland	German	German
	umcpxgg2	University of Maryland	German	German
*	umcpxgg3	University of Maryland	German	German
	umcpxgg4	University of Maryland	German	German
	umcpxgg5	University of Maryland	German	German
	umcpxgg6	University of Maryland	German	German

*Evaluation output included in the Proceedings. All runs, including the * runs, are available on the TREC Web Site (http://trec.nist.gov/pubs/trec6/t6_proceedings.html).

CROSS-LINGUAL ENGLISH RUNS

	Tag	Organization	Topic Language	Document Language
*	CEAef	CEA/DIST/SMTI	English	French
	Cor6EFent	Cornell University	English	French
*	Cor6EFexp	Cornell University	English	French
	Cor6ETGsc	Cornell University	English	German(Trans)
	97lsiLEF	Duke/U.Colorado/Bellcore	English	French
*	97lsiLEG	Duke/U.Colorado/Bellcore	English	German
	97lsiSEF	Duke/U.Colorado/Bellcore	English	French
	97lsiSEG	Duke/U.Colorado/Bellcore	English	German
	clcrl2	New Mexico State University	English	French
*	clcrl3	New Mexico State University	English	French
	clcrl4	New Mexico State University	English	French
*	ETHed1	Swiss Federal Institute of Technology (ETH)	English	German
	ETHed2	Swiss Federal Institute of Technology (ETH)	English	German
	ETHed3	Swiss Federal Institute of Technology (ETH)	English	German
	ETHed4	Swiss Federal Institute of Technology (ETH)	English	German
*	BrklyE2GA	University of California, Berkeley	English	German
	BrklyE2GM	University of California, Berkeley	English	German
*	umcpxeg1	University of Maryland	English	German
	umcpxeg2	University of Maryland	English	German
	umcpxeg3	University of Maryland	English	German
*	XRCECLE2FM	Xerox Research Centre Europe	English	French
	XRCECLE2FA	Xerox Research Centre Europe	English	French

CROSS-LINGUAL FRENCH RUNS

	Tag	Organization	Topic Language	Document Language
	97lsiLFE	Duke/U.Colorado/Bellcore	French	English
	97lsiSFE	Duke/U.Colorado/Bellcore	French	English
*	TNOfe1	TwentyOne Consortium	French	English
	TNOfe2	TwentyOne Consortium	French	English
	TNOfe3	TwentyOne Consortium	French	English
	TNOfe4	TwentyOne Consortium	French	English
	TNOfe5	TwentyOne Consortium	French	English
	TNOfe6	TwentyOne Consortium	French	English
	XRCECLF2EA	Xerox Research Centre Europe	French	English
*	XRCECLF2EM	Xerox Research Centre Europe	French	English
*	97lsiLFG	Duke/U.Colorado/Bellcore	French	German
	97lsiSFG	Duke/U.Colorado/Bellcore	French	German
*	ETHfd1	Swiss Federal Institute of Technology (ETH)	French	German
	ETHfd2	Swiss Federal Institute of Technology (ETH)	French	German

*Evaluation output included in the Proceedings. All runs, including the * runs, are available on the TREC Web Site (http://trec.nist.gov/pubs/trec6/t6_proceedings.html).

CROSS-LINGUAL GERMAN RUNS

	Tag	Organization	Topic Language	Document Language
	97lsiLGE	Duke/U.Colorado/Bellcore	German	English
*	97lsiLGF	Duke/U.Colorado/Bellcore	German	French
	97lsiSGE	Duke/U.Colorado/Bellcore	German	English
	97lsiSGF	Duke/U.Colorado/Bellcore	German	French
*	ETHde1	Swiss Federal Institute of Technology (ETH)	German	English
	ETHde2	Swiss Federal Institute of Technology (ETH)	German	English
	ETHde3	Swiss Federal Institute of Technology (ETH)	German	English
	ETHdf1	Swiss Federal Institute of Technology (ETH)	German	French
	ETHdf2	Swiss Federal Institute of Technology (ETH)	German	French
*	TNOde1	TwentyOne Consortium	German	English
	TNOde2	TwentyOne Consortium	German	English
	TNOde3	TwentyOne Consortium	German	English
	TNOde4	TwentyOne Consortium	German	English
	TNOde5	TwentyOne Consortium	German	English
	TNOde6	TwentyOne Consortium	German	English
	TNOdeMT1	TwentyOne Consortium	German	English(Trans)
*	umcpxge1	University of Maryland	German	English
	umcpxge2	University of Maryland	German	English
	umcpxge3	University of Maryland	German	English

CROSS-LINGUAL DUTCH RUNS

	Tag	Organization	Topic Language	Document Language
*	TNOnle1	TwentyOne Consortium	Dutch	English
	TNOnle2	TwentyOne Consortium	Dutch	English
	TNOnle3	TwentyOne Consortium	Dutch	English
	TNOnle4	TwentyOne Consortium	Dutch	English
	TNOnle5	TwentyOne Consortium	Dutch	English
	TNOnle6	TwentyOne Consortium	Dutch	English

CROSS-LINGUAL SPANISH RUNS

	Tag	Organization	Topic Language	Document Language
*	INQ4xl1	University of Massachusetts, Amherst	Spanish	English
	INQ4xl2	University of Massachusetts, Amherst	Spanish	English

*Evaluation output included in the Proceedings. All runs, including the * runs, are available on the TREC Web Site (http://trec.nist.gov/pubs/trec6/t6_proceedings.html).

FILTERING

Tag	Organization
att97fcasp	AT&T Labs Research
att97fcrank	AT&T Labs Research
att97fcuf1	AT&T Labs Research
att97fcuf2	AT&T Labs Research
att97feasp	AT&T Labs Research
att97ferank	AT&T Labs Research
att97feuf1	AT&T Labs Research
att97feuf2	AT&T Labs Research
anu6fltU1	Australian National University
anu6fltU2	Australian National University
city6f11	City University
city6f12	City University
city6f13	City University
city6f21	City University
city6f22	City University
city6f23	City University
CLComm	CLARITECH Corporation
CLCommASP	CLARITECH Corporation
CLCommF1	CLARITECH Corporation
CLCommF2	CLARITECH Corporation
CLRoute	CLARITECH Corporation
CLRouteASP	CLARITECH Corporation
CLRouteF1	CLARITECH Corporation
CLRouteF2	CLARITECH Corporation
dbulm1Asp	Daimler Benz Research Center Ulm
dbulm1F1R	Daimler Benz Research Center Ulm
dbulm1fF1	Daimler Benz Research Center Ulm
dbulm1fF2	Daimler Benz Research Center Ulm
dbulm1fF2R	Daimler Benz Research Center Ulm
pir7fa1	Queens College, CUNY
pir7fa2	Queens College, CUNY
pir7f11	Queens College, CUNY
pir7f12	Queens College, CUNY
pir7f21	Queens College, CUNY
pir7f22	Queens College, CUNY
teklis6	Siemens AG
teklis7	Siemens AG
teklis65	Siemens AG
teklis75	Siemens AG
teklis1000	Siemens AG

FILTERING (CONTINUED)

Tag	Organization
BKYT6fASP1	University of California, Berkeley
BKYT6fBOOL1	University of California, Berkeley
BKYT6fF11	University of California, Berkeley
BKYT6fF21	University of California, Berkeley
BKYT6fRANK1	University of California, Berkeley
INQ415	University of Massachusetts, Amherst
INQ416	University of Massachusetts, Amherst
INQ417	University of Massachusetts, Amherst
INQ418	University of Massachusetts, Amherst
INQ419	University of Massachusetts, Amherst
INQ420	University of Massachusetts, Amherst
INQ421	University of Massachusetts, Amherst
INQ422	University of Massachusetts, Amherst
INQ423	University of Massachusetts, Amherst
INQ424	University of Massachusetts, Amherst
isf1	University of North Carolina (Newby)
isf1r	University of North Carolina (Newby)
isf2	University of North Carolina (Newby)
isf2r	University of North Carolina (Newby)

HIGH PRECISION

Tag	Organization
Cor6HP1	Cornell University
Cor6HP2	Cornell University
Cor6HP3	Cornell University
DCU97HP	Dublin City University
otc1	Open Text Corporation
otc2	Open Text Corporation
otc3	Open Text Corporation
pirc7Ha	Queens College, CUNY
pirc7Hd	Queens College, CUNY
pirc7Ht	Queens College, CUNY
uwmt6h0	University of Waterloo
uwmt6h1	University of Waterloo
uwmt6h2	University of Waterloo

INTERACTIVE

Tag	Organization
city	City University
ibm	IBM T.J. Watson Research Center (Schmidt-Wesche)
nmsu	New Mexico State University (Ogden)
ohsu	Oregon Health Sciences University
rmit	RMIT
rutgers	Rutgers University (Belkin)
berkeley	University of California, Berkeley
umass	University of Massachusetts, Amherst
unc	University of North Carolina (Sumner)

NLP

Tag	Organization
genlp1	GE/Rutgers/Lockheed Martin/SICS
genlp2	GE/Rutgers/Lockheed Martin/SICS
genlp3	GE/Rutgers/Lockheed Martin/SICS
Gla6DS1	University of Glasgow
Gla6DS2	University of Glasgow
Gla6DS3	University of Glasgow

SDR

Tag	Organization
att97sB1	AT&T Labs Research
att97sR1	AT&T Labs Research
att97sS1	AT&T Labs Research
att97sS2	AT&T Labs Research
CMUcmu	Carnegie Mellon University
CMUibm	Carnegie Mellon University
CMUref	Carnegie Mellon University
citysdrB1	City University
citysdrB2	City University
citysdrR1	City University
citysdrR2	City University
CLARITsdrB1	CLARITECH Corporation
CLARITsdrB2	CLARITECH Corporation
CLARITsdrR1	CLARITECH Corporation
CLARITsdrS1	CLARITECH Corporation
CLARITsdrS2	CLARITECH Corporation
DCU97QSDRB1	Dublin City University
DCU97QSDRB2	Dublin City University
DCU97QSDRR1	Dublin City University
DCU97QSDRR2	Dublin City University
DCU97rest	Dublin City University
ibms97s	IBM T.J. Watson Research Center (Roukos)
ibms97t	IBM T.J. Watson Research Center (Roukos)
mds612	RMIT
mds613	RMIT
mds614	RMIT
mds615	RMIT
ETHB1	Swiss Federal Institute of Technology
ETHB2	Swiss Federal Institute of Technology
ETHR1	Swiss Federal Institute of Technology
ETHS1	Swiss Federal Institute of Technology
ETHS2	Swiss Federal Institute of Technology
gla6B1	University of Glasgow
gla6R1	University of Glasgow
gla6S1	University of Glasgow
gla6S2	University of Glasgow

SDR (CONTINUED)

Tag	Organization
umcp97l1	University of Maryland
umcp97s2	University of Maryland
INQ4sdd	University of Massachusetts, Amherst
INQ4sdl	University of Massachusetts, Amherst
INQ4sds	University of Massachusetts, Amherst
THISLB1	University of Sheffield
THISLB2	University of Sheffield
THISLR1	University of Sheffield
THISLR2	University of Sheffield
THISLS1	University of Sheffield
THISLS2	University of Sheffield
nsasglt1	U.S. Department of Defense
nsasgsr1	U.S. Department of Defense

Evaluation Techniques and Measures

Categories

The results following this section are organized according to the task accomplished by the run: ad hoc, routing, or a track task.

I. Ad hoc
Retrieval using an "ad hoc" topic such as a researcher might use in a library environment. In TREC this implies that the input topic has no training material such as relevance judgments to aid in the construction of the input query.

A. Category A
Systems running TREC topics against all documents from TREC Disks 4 and 5.

B. Category B
Systems running TREC topics against the Financial Times data on TREC Disk 4. (Intended for new groups, allowing them to scale their systems to handle large collections.)

II. Routing
Retrieval using a "routing" query such as a profile to filter some incoming document stream. In TREC this implies that the input topic has training material, including relevance judgments against the training documents, to use in constructing the input query or profile. This query is then used against new documents (the test documents).

A. Category A
Systems running TREC topics against a set of Foreign Broadcast Information Service (FBIS) documents.

B. Category B
Systems running TREC topics against FBIS documents contained in files fb6-f001 through fb6-f225. (Intended for new groups, allowing them to scale their systems to handle large collections.)

Evaluation Measures

I. Recall
A measure of the ability of a system to present all relevant items.

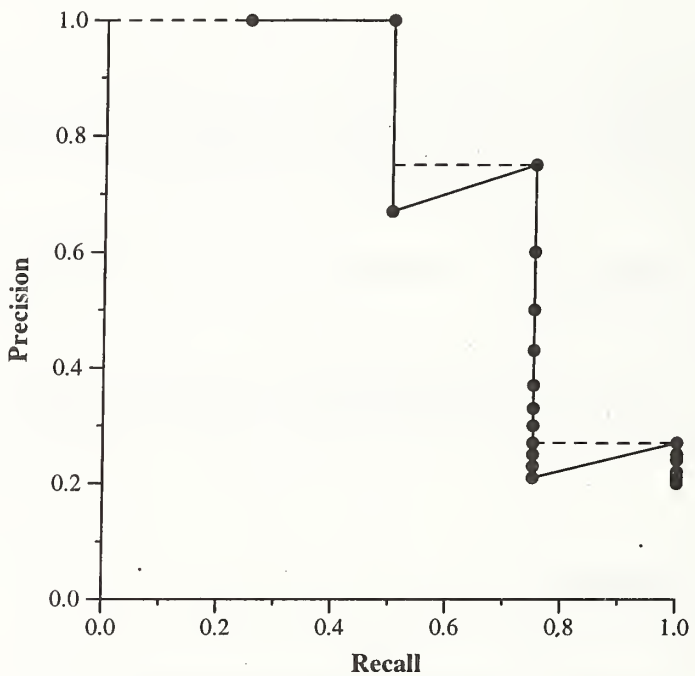
$$\text{recall} = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}}$$

II. Precision.
A measure of the ability of a system to present only relevant items.

$$\text{precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}}$$

Precision and recall are set-based measures. That is, they evaluate the quality of an unordered set of retrieved documents. To evaluate ranked lists, precision can be plotted against recall after each retrieved document as shown in the example below. To facilitate computing average performance over a set of topics, each with a different number of relevant documents, individual topic precision values are interpolated to a set of standard recall levels (0 to 1 in increments of .1). The particular rule used to interpolate precision at standard recall level i is to use the maximum precision obtained for the topic for any actual recall level greater than or equal to i. Note that while precision is not defined at a recall of 0.0, this interpolation rule does define an interpolated value for recall level 0.0. In the example, the actual precision values are plotted with circles (and connected by a solid line) and the interpolated precision is shown with the dashed line.

Example: Assume a document collection has 20 documents, four of which are relevant to topic t. Further assume a retrieval system ranks the relevant documents first, second, fourth, and fifteenth. The exact recall points are 0.25, 0.5, 0.75, and 1.0. Using the interpolation rule, the interpolated precision for all standard recall levels up to .5 is 1, the interpolated precision for recall levels .6 and .7 is .75, and the interpolated precision for recall levels .8 or greater is .27.



System Results Description

Each of the following pages contains the evaluation results for one run. A page is comprised of a header (containing the task and organization name), 3 tables, and 2 graphs.

Tables

Tables are generated by *trec_eval* courtesy of Chris Buckley using the SMART methodology.

I. "Summary Statistics" Table
Table 1 is a sample "Summary Statistics" Table

Table 1: Sample "Summary Statistics" Table.

Summary Statistics	
Run	Cor5A2cr–category A, automatic, short topic
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	5524
Rel_ret:	2848

A. Run
A description of the run. It contains the run tag provided by the participant, and as applicable, whether the run is Category A or B, whether queries were constructed manually or automatically, and whether long or short topic descriptions were used.

B. Number of Topics
Number of topics searched in this run (generally 50 topics are run for each task).

C. Total number of documents over all topics (the number of topics given in B).

i. Retrieved
Number of documents submitted to NIST. This is usually 50,000 (50 topics $\times$ 1000 documents), but is less when fewer than 1000 documents are retrieved per topic.

ii. Relevant
Total possible relevant documents within a given task and category.

iii. Rel_ret
Total number of relevant documents returned by a run over all the topics.

II. "Recall Level Precision Averages" Table.
Table 2 is a sample "Recall Level Precision Averages" Table.

A. Precision at 11 standard recall levels
The precision averages at 11 standard recall levels are used to compare the performance of different systems and as the input for plotting the recall-precision graph (see below). Each recall-precision average is computed by summing the interpolated precisions at the specified recall cutoff value (denoted by $\sum P_\lambda$ where P_λ is the interpolated precision at

Table 2: Sample "Recall Level Precision Averages" Table.

Recall Level Precision Averages	
Recall	Precision
0.00	0.5857
0.10	0.3927
0.20	0.3252
0.30	0.2799
0.40	0.2521
0.50	0.2131
0.60	0.1776
0.70	0.1395
0.80	0.0885
0.90	0.0415
1.00	0.0118
Average precision over all relevant docs	
non-interpolated	0.2109

recall level λ) and then dividing by the number of topics.

$$\frac{\sum_{i=1}^{NUM} P_\lambda}{NUM} \qquad \lambda = \{0.0, 0.1, 0.2, 0.3, \ldots, 1.0\}$$

- Interpolating recall-precision
  Standard recall levels facilitate averaging and plotting retrieval results.

B. Average precision over all relevant documents, non-interpolated
This is a single-valued measure that reflects the performance over all relevant documents. It rewards systems that retrieve relevant documents quickly (highly ranked).
The measure is not an average of the precision at standard recall levels. Rather, it is the average of the precision value obtained after each relevant document is retrieved. (When a relevant document is not retrieved at all, its precision is assumed to be 0.) As an example, consider a query that has four relevant documents which are retrieved at ranks 1, 2, 4, and 7. The actual precision obtained when each relevant document is retrieved is 1, 1, 0.75, and 0.57, respectively, the mean of which is 0.83. Thus, the average precision over all relevant documents for this query is 0.83.

III. "Document Level Averages" Table
Table 3 is a sample "Document Level Averages" Table.

A. Precision at 9 document cutoff values
The precision computed after a given number of documents have been retrieved reflects the actual measured system performance as a user might see it. Each document precision average is computed by summing the precisions at the specified document cutoff value and dividing by the number of topics (50).

B. R-Precision
R-Precision is the precision after R documents have been retrieved, where R is the

Table 3: Sample "Document Level Averages" Table.

Document Level Averages	
	Precision
At 5 docs	0.4240
At 10 docs	0.3800
At 15 docs	0.3453
At 20 docs	0.3270
At 30 docs	0.2913
At 100 docs	0.2018
At 200 docs	0.1544
At 500 docs	0.0933
At 1000 docs	0.0570
R−Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2404

number of relevant documents for the topic. It de-emphasizes the exact ranking of the retrieved relevant documents, which can be particularly useful in TREC where there are large numbers of relevant documents.

The average R-Precision for a run is computed by taking the mean of the R-Precisions of the individual topics in the run. For example, assume a run consists of two topics, one with 50 relevant documents and another with 10 relevant documents. If the retrieval system returns 17 relevant documents in the top 50 documents for the first topic, and 7 relevant documents in the top 10 for the second topic, then the run's R-Precision would be $\frac{\frac{17}{50} + \frac{7}{10}}{2}$ or 0.52.

Graphs

I. Recall-Precision Graph
Figure 1 is a sample Recall-Precision Graph.

The Recall-Precision Graph is created using the 11 cutoff values from the Recall Level Precision Averages. Typically these graphs slope downward from left to right, enforcing the notion that as more relevant documents are retrieved (recall increases), the more nonrelevant documents are retrieved (precision decreases).

This graph is the most commonly used method for comparing systems. The plots of different runs can be superimposed on the same graph to determine which run is superior. Curves closest to the upper right-hand corner of the graph (where recall and precision are maximized) indicate the best performance. Comparisons are best made in three different recall ranges: 0 to 0.2, 0.2 to 0.8, and 0.8 to 1. These ranges characterize high precision, middle recall, and high recall performance, respectively.

II. Average Precision Histogram.
Figure 2 is a sample Average Precision Histogram.

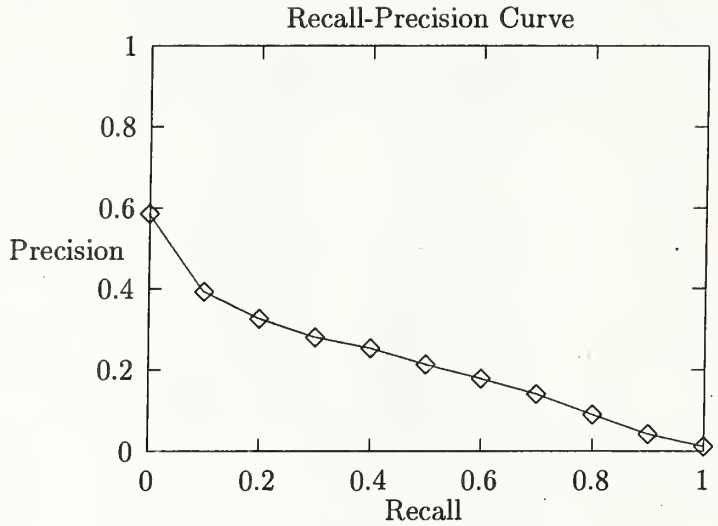


Figure 1: Sample Recall-Precision Graph.

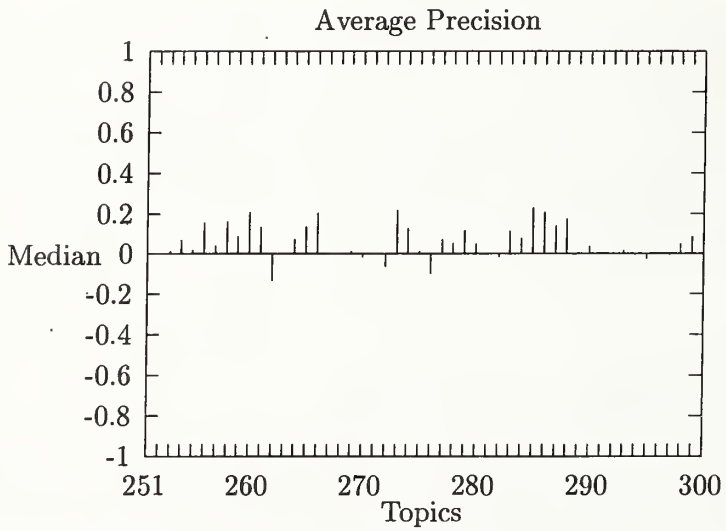


Figure 2: Sample Average Precision Histogram.

The Average Precision Histogram measures the average precision of a run on each topic against the median average precision of all corresponding runs on that topic. This graph is intended to give insight into the performance of individual systems and the types of topics that they handle well.

Summary Statistics	
Run Number	aiatA1
Run Description	Category A, Automatic, short
Number of Topics	50

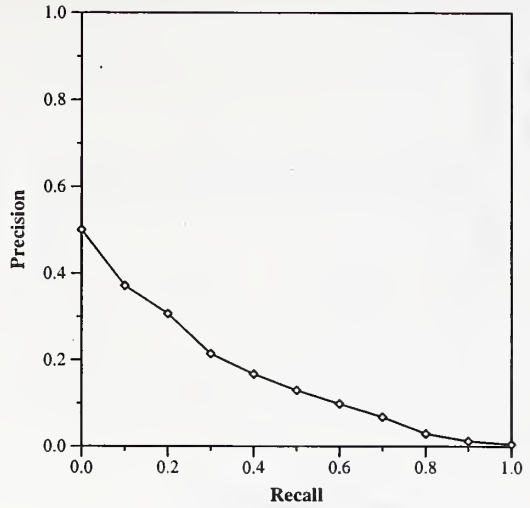
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1905

Recall Level Precision Averages	
Recall	Precision
0.00	0.5005
0.10	0.3720
0.20	0.3073
0.30	0.2145
0.40	0.1676
0.50	0.1298
0.60	0.0987
0.70	0.0681
0.80	0.0297
0.90	0.0124
1.00	0.0048

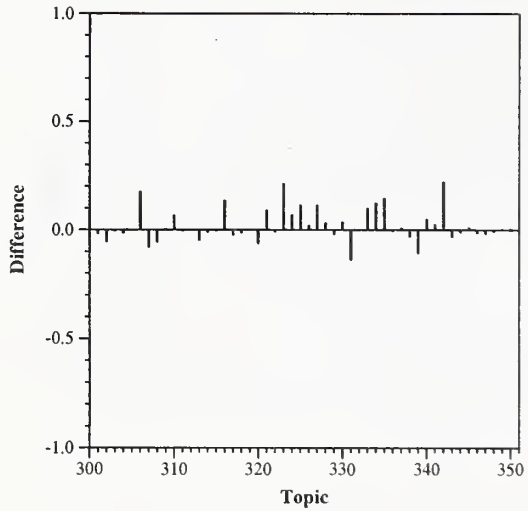
Average precision over all relevant docs	
non-interpolated	0.1566

Document Level Averages	
	Precision
At 5 docs	0.3440
At 10 docs	0.3080
At 15 docs	0.2653
At 20 docs	0.2480
At 30 docs	0.2220
At 100 docs	0.1402
At 200 docs	0.0988
At 500 docs	0.0578
At 1000 docs	0.0381

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1878



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	aiatB1
Run Description	Category A, Automatic, title
Number of Topics	50

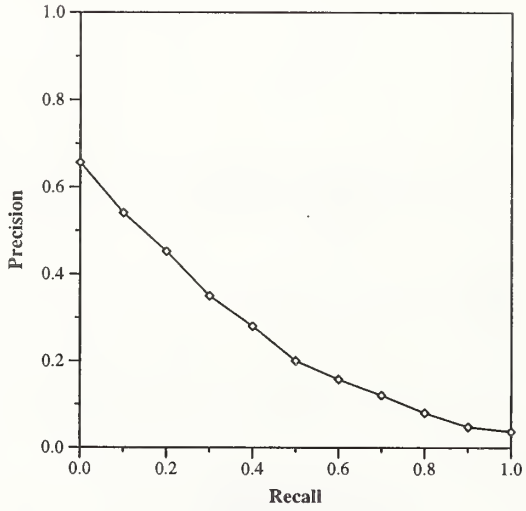
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2475

Recall Level Precision Averages	
Recall	Precision
0.00	0.6566
0.10	0.5409
0.20	0.4526
0.30	0.3505
0.40	0.2805
0.50	0.2008
0.60	0.1580
0.70	0.1212
0.80	0.0806
0.90	0.0483
1.00	0.0376

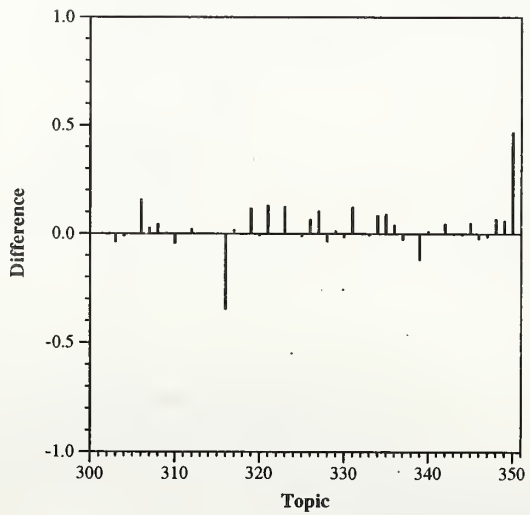
Average precision over all relevant docs	
non-interpolated	0.2481

Document Level Averages	
	Precision
At 5 docs	0.4920
At 10 docs	0.4320
At 15 docs	0.4040
At 20 docs	0.3740
At 30 docs	0.3360
At 100 docs	0.2104
At 200 docs	0.1446
At 500 docs	0.0800
At 1000 docs	0.0495

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2808



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	att97ac
Run Description	Category A, Automatic, short
Number of Topics	50

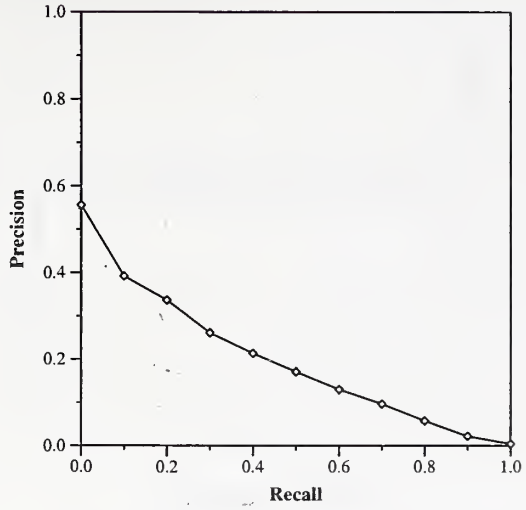
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2421

Recall Level Precision Averages	
Recall	Precision
0.00	0.5561
0.10	0.3923
0.20	0.3369
0.30	0.2613
0.40	0.2139
0.50	0.1712
0.60	0.1304
0.70	0.0968
0.80	0.0578
0.90	0.0219
1.00	0.0043

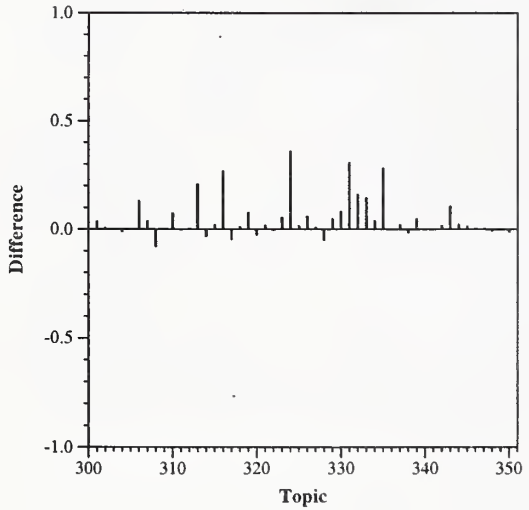
Average precision over all relevant docs	
non-interpolated	0.1847

Document Level Averages	
	Precision
At 5 docs	0.3560
At 10 docs	0.3260
At 15 docs	0.3027
At 20 docs	0.2770
At 30 docs	0.2553
At 100 docs	0.1822
At 200 docs	0.1309
At 500 docs	0.0768
At 1000 docs	0.0484

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2241



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	att97ae
Run Description	Category A, Automatic, short
Number of Topics	50

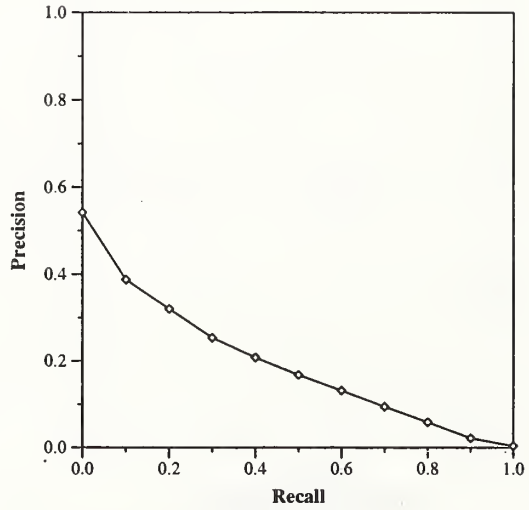
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2421

Recall Level Precision Averages	
Recall	Precision
0.00	0.5419
0.10	0.3877
0.20	0.3205
0.30	0.2539
0.40	0.2083
0.50	0.1680
0.60	0.1320
0.70	0.0949
0.80	0.0589
0.90	0.0220
1.00	0.0040

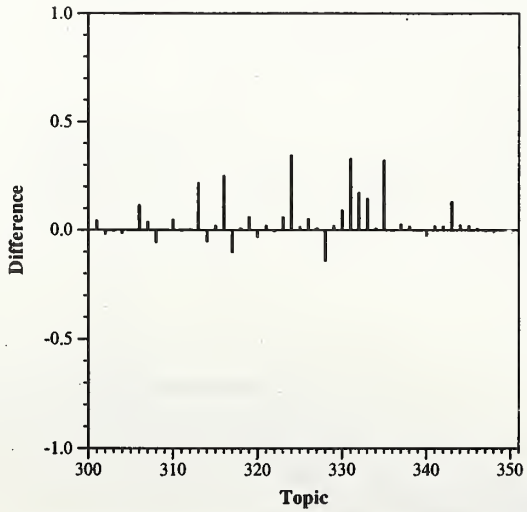
Average precision over all relevant docs	
non-interpolated	0.1801

Document Level Averages	
	Precision
At 5 docs	0.3520
At 10 docs	0.3220
At 15 docs	0.2947
At 20 docs	0.2800
At 30 docs	0.2533
At 100 docs	0.1794
At 200 docs	0.1303
At 500 docs	0.0775
At 1000 docs	0.0484

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2203



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	att97as
Run Description	Category A, Automatic, title
Number of Topics	50

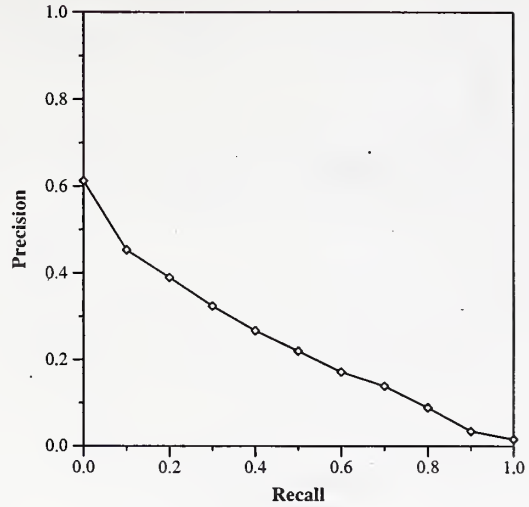
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2516

Recall Level Precision Averages	
Recall	Precision
0.00	0.6126
0.10	0.4536
0.20	0.3901
0.30	0.3241
0.40	0.2672
0.50	0.2201
0.60	0.1717
0.70	0.1390
0.80	0.0893
0.90	0.0347
1.00	0.0159

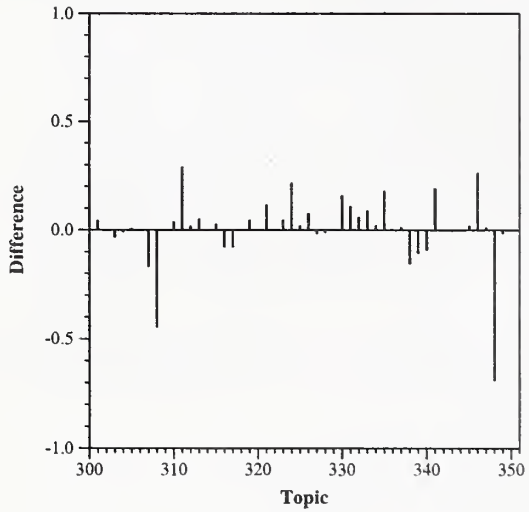
Average precision over all relevant docs	
non-interpolated	0.2289

Document Level Averages	
	Precision
At 5 docs	0.4240
At 10 docs	0.3900
At 15 docs	0.3600
At 20 docs	0.3530
At 30 docs	0.3233
At 100 docs	0.2164
At 200 docs	0.1472
At 500 docs	0.0830
At 1000 docs	0.0503

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2592



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	anu6alo1
Run Description	Category A, Automatic, long
Number of Topics	50

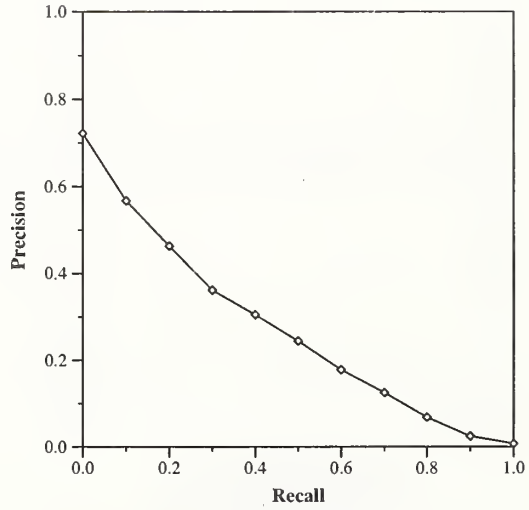
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2844

Recall Level Precision Averages	
Recall	Precision
0.00	0.7224
0.10	0.5675
0.20	0.4633
0.30	0.3623
0.40	0.3054
0.50	0.2448
0.60	0.1785
0.70	0.1254
0.80	0.0685
0.90	0.0247
1.00	0.0073

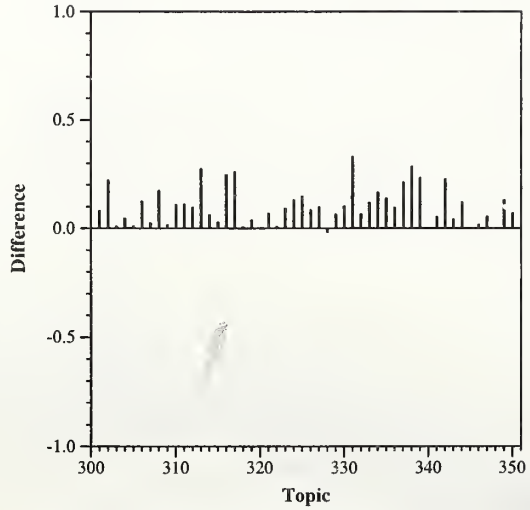
Average precision over all relevant docs	
non-interpolated	0.2602

Document Level Averages	
	Precision
At 5 docs	0.5400
At 10 docs	0.4480
At 15 docs	0.4027
At 20 docs	0.3790
At 30 docs	0.3360
At 100 docs	0.2238
At 200 docs	0.1586
At 500 docs	0.0910
At 1000 docs	0.0569

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2995



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	anu6ash1
Run Description	Category A, Automatic, short
Number of Topics	50

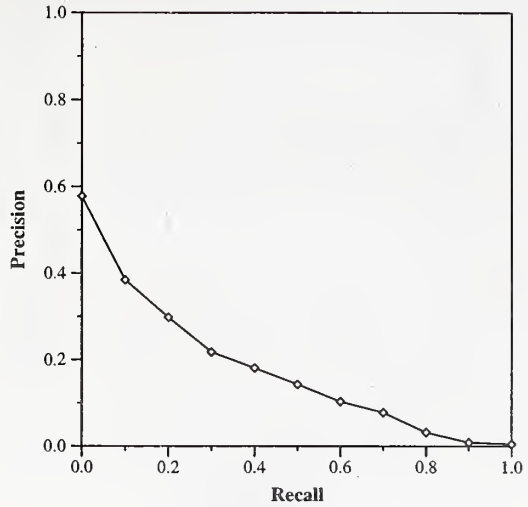
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2213

Recall Level Precision Averages	
Recall	Precision
0.00	0.5780
0.10	0.3851
0.20	0.2987
0.30	0.2185
0.40	0.1812
0.50	0.1436
0.60	0.1036
0.70	0.0781
0.80	0.0321
0.90	0.0090
1.00	0.0051

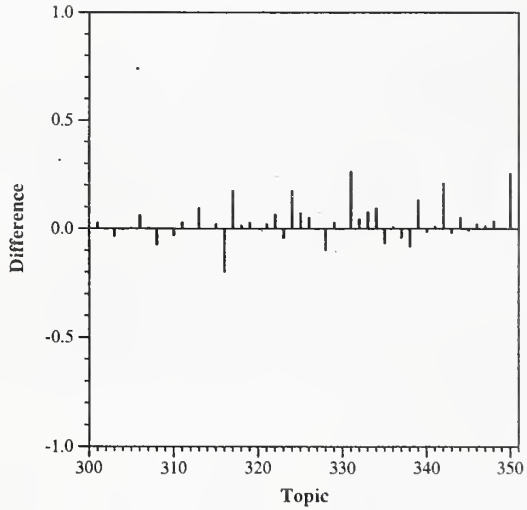
Average precision over all relevant docs	
non-interpolated	0.1645

Document Level Averages	
	Precision
At 5 docs	0.3680
At 10 docs	0.3220
At 15 docs	0.3067
At 20 docs	0.2820
At 30 docs	0.2507
At 100 docs	0.1634
At 200 docs	0.1164
At 500 docs	0.0674
At 1000 docs	0.0443

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1887



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	city6ad
Run Description	Category A, Automatic, short
Number of Topics	50

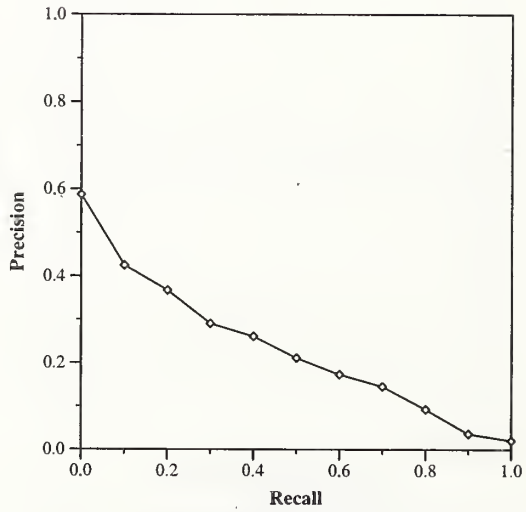
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1965

Recall Level Precision Averages	
Recall	Precision
0.00	0.5870
0.10	0.4247
0.20	0.3676
0.30	0.2909
0.40	0.2611
0.50	0.2115
0.60	0.1735
0.70	0.1457
0.80	0.0929
0.90	0.0365
1.00	0.0209

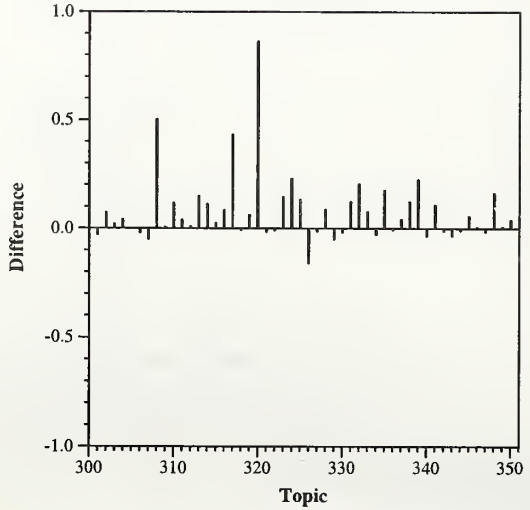
Average precision over all relevant docs	
non-interpolated	0.2164

Document Level Averages	
	Precision
At 5 docs	0.4200
At 10 docs	0.3560
At 15 docs	0.3093
At 20 docs	0.2830
At 30 docs	0.2620
At 100 docs	0.1666
At 200 docs	0.1131
At 500 docs	0.0623
At 1000 docs	0.0393

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2498



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	city6al
Run Description	Category A, Automatic, long
Number of Topics	50

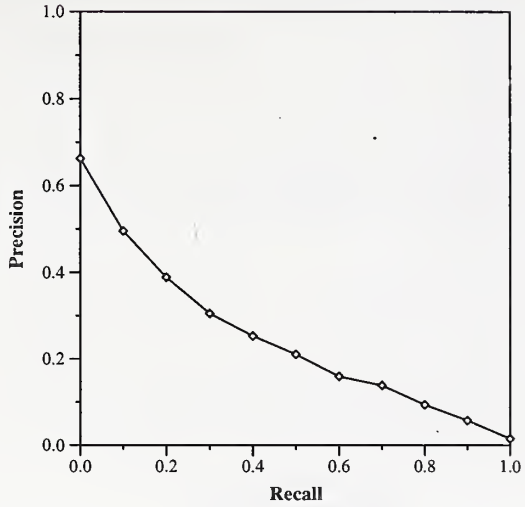
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2422

Recall Level Precision Averages	
Recall	Precision
0.00	0.6630
0.10	0.4959
0.20	0.3892
0.30	0.3054
0.40	0.2533
0.50	0.2107
0.60	0.1595
0.70	0.1387
0.80	0.0940
0.90	0.0577
1.00	0.0153

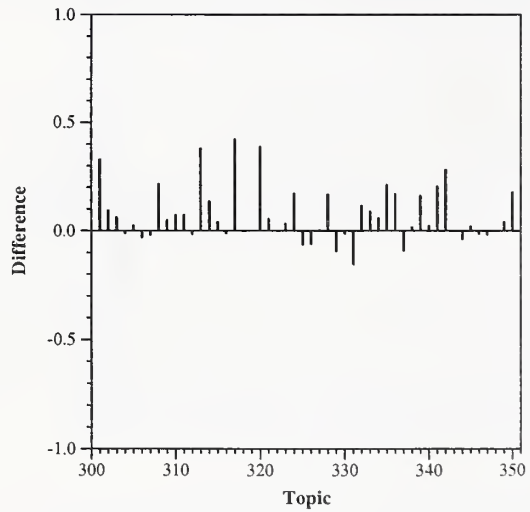
Average precision over all relevant docs	
non-interpolated	0.2327

Document Level Averages	
	Precision
At 5 docs	0.4360
At 10 docs	0.3940
At 15 docs	0.3573
At 20 docs	0.3320
At 30 docs	0.2867
At 100 docs	0.1934
At 200 docs	0.1347
At 500 docs	0.0768
At 1000 docs	0.0484

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2595



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	city6at
Run Description	Category A, Automatic, title
Number of Topics	50

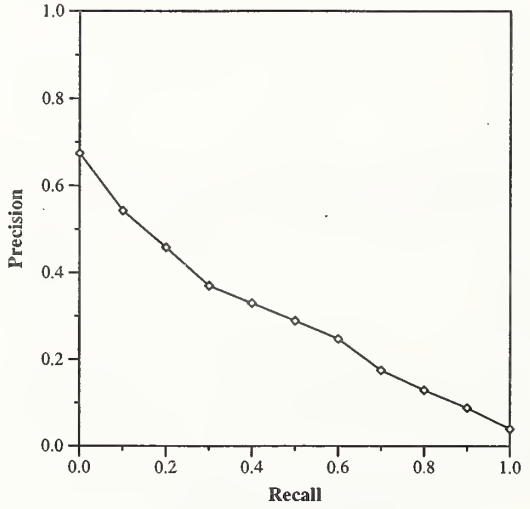
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2560

Recall Level Precision Averages	
Recall	Precision
0.00	0.6745
0.10	0.5428
0.20	0.4586
0.30	0.3705
0.40	0.3300
0.50	0.2894
0.60	0.2481
0.70	0.1755
0.80	0.1297
0.90	0.0885
1.00	0.0400

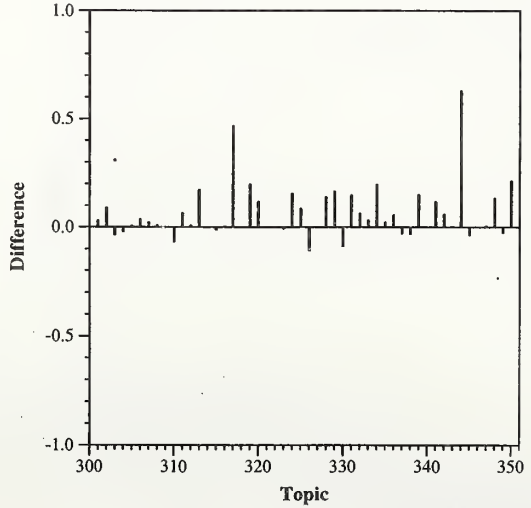
Average precision over all relevant docs	
non-interpolated	0.2876

Document Level Averages	
	Precision
At 5 docs	0.4800
At 10 docs	0.4380
At 15 docs	0.3933
At 20 docs	0.3670
At 30 docs	0.3200
At 100 docs	0.2196
At 200 docs	0.1542
At 500 docs	0.0852
At 1000 docs	0.0512

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3220



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Cor6A1cls
Run Description	Category A, Automatic, short
Number of Topics	50

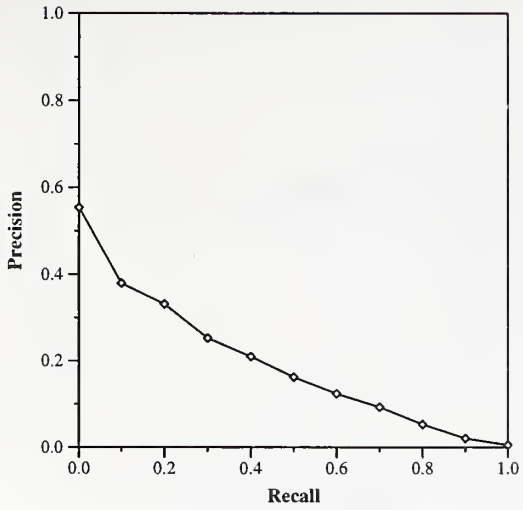
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2391

Recall Level Precision Averages	
Recall	Precision
0.00	0.5541
0.10	0.3795
0.20	0.3314
0.30	0.2527
0.40	0.2100
0.50	0.1625
0.60	0.1241
0.70	0.0933
0.80	0.0534
0.90	0.0209
1.00	0.0054

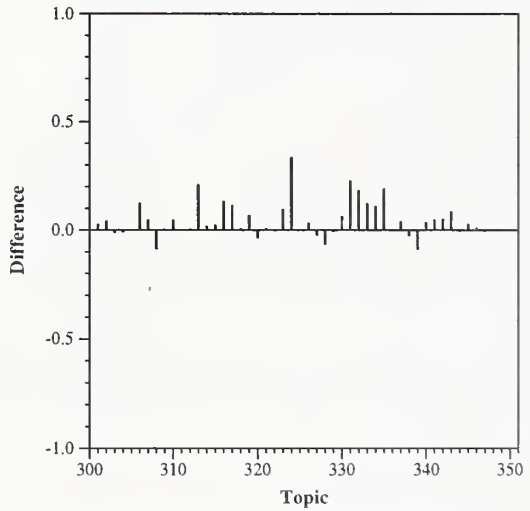
Average precision over all relevant docs	
non-interpolated	0.1799

Document Level Averages	
	Precision
At 5 docs	0.3600
At 10 docs	0.3100
At 15 docs	0.3013
At 20 docs	0.2800
At 30 docs	0.2560
At 100 docs	0.1794
At 200 docs	0.1308
At 500 docs	0.0765
At 1000 docs	0.0478

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2155



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Cor6A2qtcs
Run Description	Category A, Automatic, short
Number of Topics	50

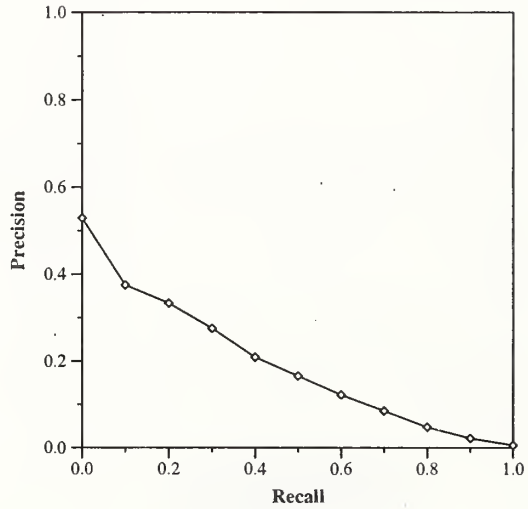
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2332

Recall Level Precision Averages	
Recall	Precision
0.00	0.5292
0.10	0.3754
0.20	0.3336
0.30	0.2754
0.40	0.2092
0.50	0.1657
0.60	0.1223
0.70	0.0853
0.80	0.0476
0.90	0.0221
1.00	0.0059

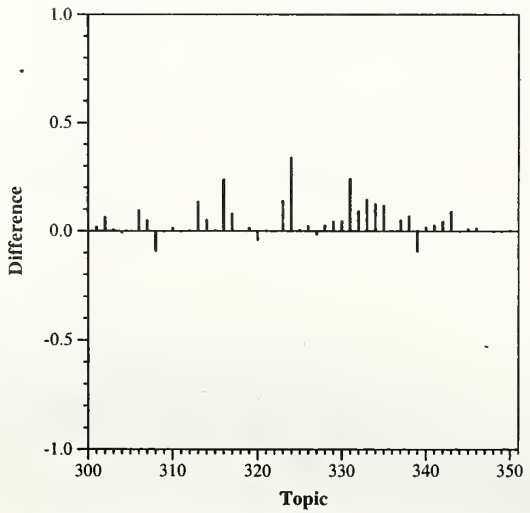
Average precision over all relevant docs	
non-interpolated	0.1809

Document Level Averages	
	Precision
At 5 docs	0.3680
At 10 docs	0.3340
At 15 docs	0.3040
At 20 docs	0.2830
At 30 docs	0.2493
At 100 docs	0.1714
At 200 docs	0.1224
At 500 docs	0.0744
At 1000 docs	0.0466

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2076



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Cor6A3cll
Run Description	Category A, Automatic, long
Number of Topics	50

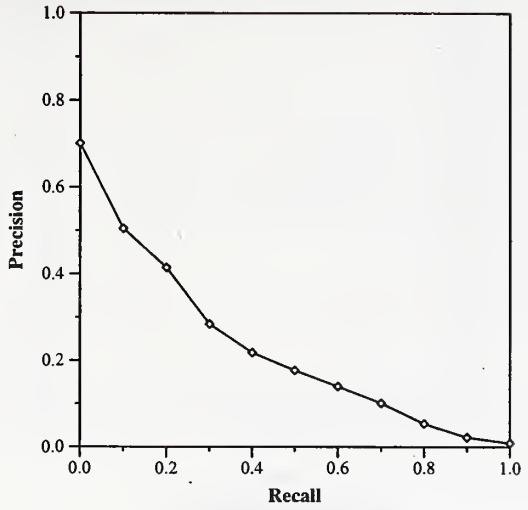
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2590

Recall Level Precision Averages	
Recall	Precision
0.00	0.7013
0.10	0.5050
0.20	0.4150
0.30	0.2846
0.40	0.2187
0.50	0.1775
0.60	0.1402
0.70	0.1015
0.80	0.0538
0.90	0.0224
1.00	0.0091

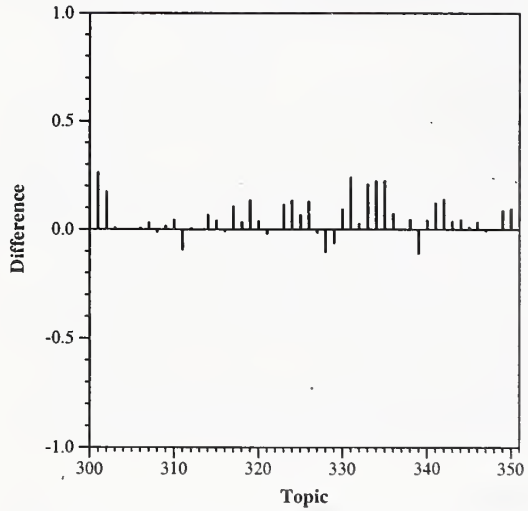
Average precision over all relevant docs	
non-interpolated	0.2139

Document Level Averages	
	Precision
At 5 docs	0.4480
At 10 docs	0.4260
At 15 docs	0.4013
At 20 docs	0.3630
At 30 docs	0.3200
At 100 docs	0.2010
At 200 docs	0.1418
At 500 docs	0.0823
At 1000 docs	0.0518

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2415



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	csiro97a1
Run Description	Category A, Automatic, long
Number of Topics	50

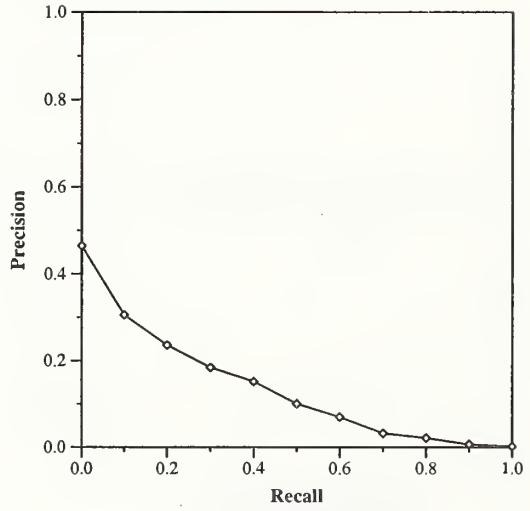
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1421

Recall Level Precision Averages	
Recall	Precision
0.00	0.4647
0.10	0.3057
0.20	0.2361
0.30	0.1844
0.40	0.1519
0.50	0.1008
0.60	0.0701
0.70	0.0327
0.80	0.0217
0.90	0.0069
1.00	0.0023

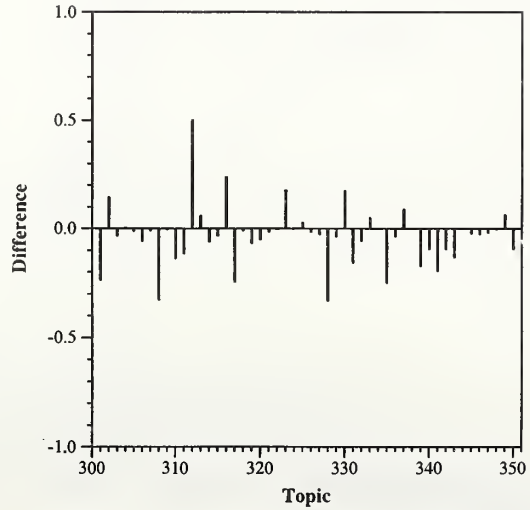
Average precision over all relevant docs	
non-interpolated	0.1265

Document Level Averages	
	Precision
At 5 docs	0.3360
At 10 docs	0.2820
At 15 docs	0.2360
At 20 docs	0.2180
At 30 docs	0.1827
At 100 docs	0.1136
At 200 docs	0.0756
At 500 docs	0.0437
At 1000 docs	0.0284

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1455



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	csiro97a2
Run Description	Category A, Automatic, short
Number of Topics	50

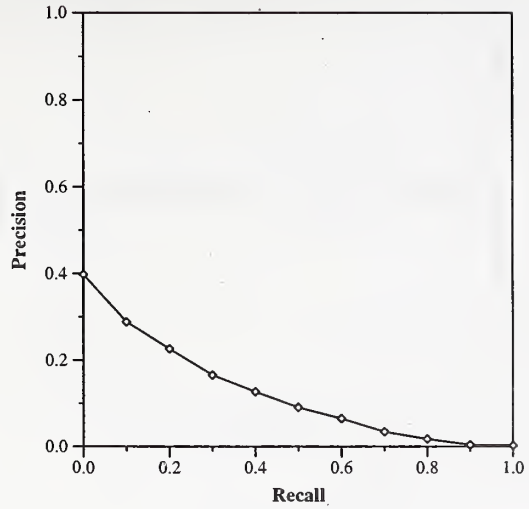
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1058

Recall Level Precision Averages	
Recall	Precision
0.00	0.3980
0.10	0.2888
0.20	0.2265
0.30	0.1657
0.40	0.1272
0.50	0.0907
0.60	0.0649
0.70	0.0345
0.80	0.0175
0.90	0.0043
1.00	0.0025

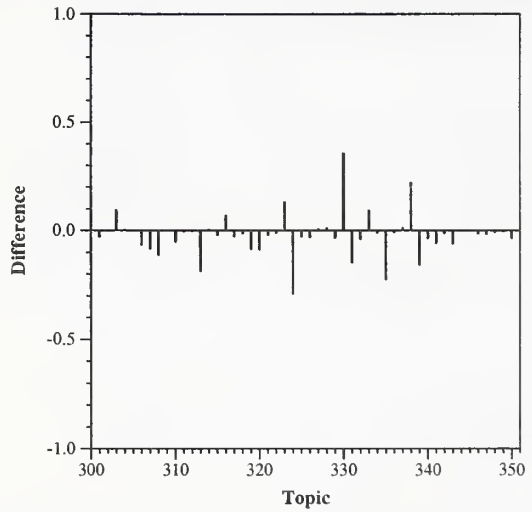
Average precision over all relevant docs	
non-interpolated	0.1171

Document Level Averages	
	Precision
At 5 docs	0.2720
At 10 docs	0.2280
At 15 docs	0.2053
At 20 docs	0.1840
At 30 docs	0.1507
At 100 docs	0.0864
At 200 docs	0.0584
At 500 docs	0.0338
At 1000 docs	0.0212

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1295



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	csiro97a3
Run Description	Category A, Automatic, title
Number of Topics	50

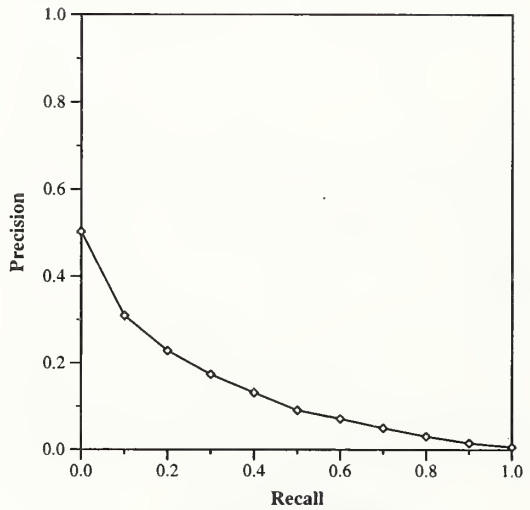
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1567

Recall Level Precision Averages	
Recall	Precision
0.00	0.5029
0.10	0.3092
0.20	0.2288
0.30	0.1739
0.40	0.1319
0.50	0.0912
0.60	0.0716
0.70	0.0503
0.80	0.0309
0.90	0.0155
1.00	0.0064

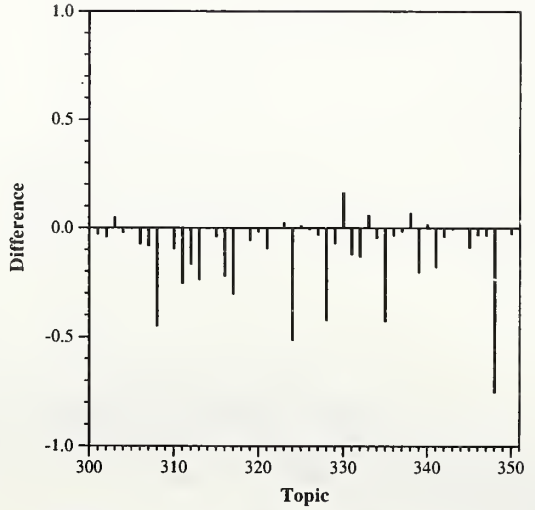
Average precision over all relevant docs	
non-interpolated	0.1259

Document Level Averages	
	Precision
At 5 docs	0.3440
At 10 docs	0.2860
At 15 docs	0.2507
At 20 docs	0.2290
At 30 docs	0.1953
At 100 docs	0.1136
At 200 docs	0.0760
At 500 docs	0.0462
At 1000 docs	0.0313

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1481



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	DCU97lnt
Run Description	Category A, Automatic, long
Number of Topics	50

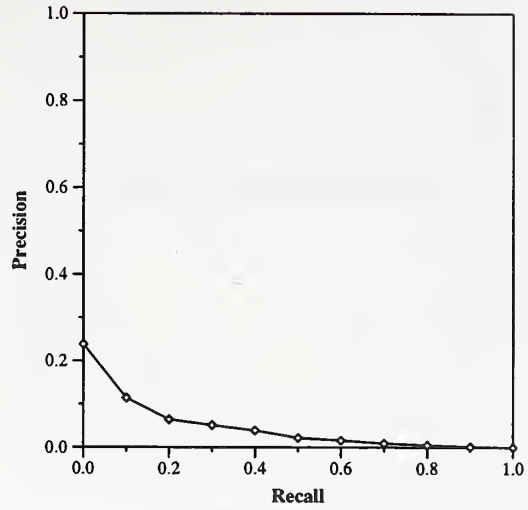
Total number of documents over all topics	
Retrieved:	50050
Relevant:	4611
Rel-ret:	1488

Recall Level Precision Averages	
Recall	Precision
0.00	0.2384
0.10	0.1143
0.20	0.0646
0.30	0.0513
0.40	0.0391
0.50	0.0222
0.60	0.0163
0.70	0.0095
0.80	0.0053
0.90	0.0016
1.00	0.0002

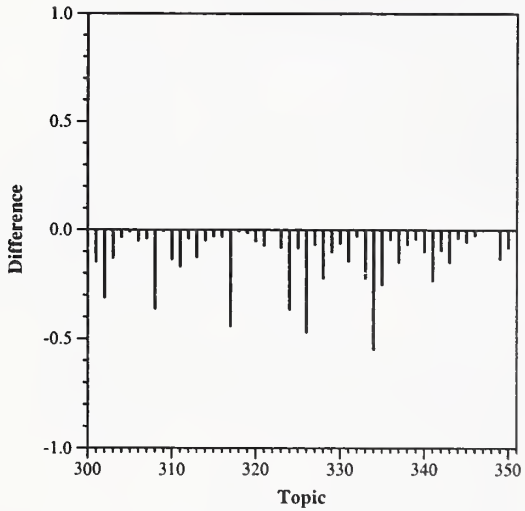
Average precision over all relevant docs	
non-interpolated	0.0372

Document Level Averages	
	Precision
At 5 docs	0.0960
At 10 docs	0.0820
At 15 docs	0.0733
At 20 docs	0.0700
At 30 docs	0.0687
At 100 docs	0.0530
At 200 docs	0.0444
At 500 docs	0.0370
At 1000 docs	0.0298

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0576



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	DCU97lt
Run Description	Category A, Automatic, long
Number of Topics	50

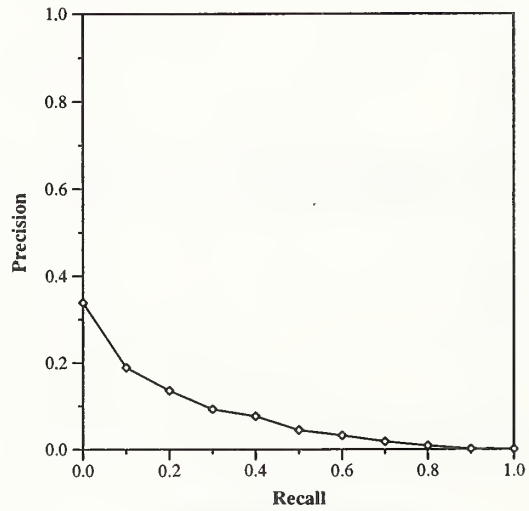
Total number of documents over all topics	
Retrieved:	50050
Relevant:	4611
Rel-ret:	1637

Recall Level Precision Averages	
Recall	Precision
0.00	0.3388
0.10	0.1895
0.20	0.1362
0.30	0.0930
0.40	0.0770
0.50	0.0448
0.60	0.0325
0.70	0.0185
0.80	0.0085
0.90	0.0015
1.00	0.0009

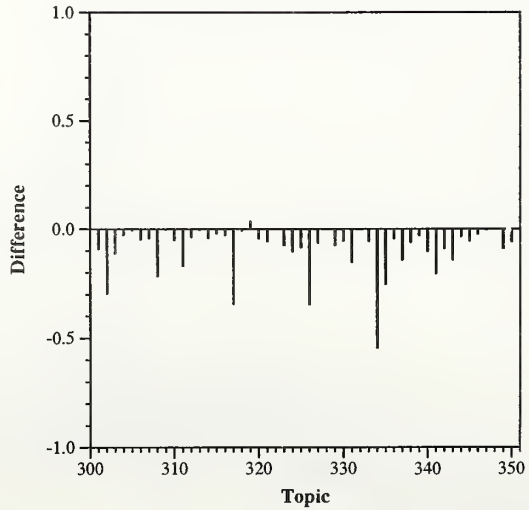
Average precision over all relevant docs	
non-interpolated	0.0696

Document Level Averages	
	Precision
At 5 docs	0.1600
At 10 docs	0.1460
At 15 docs	0.1347
At 20 docs	0.1220
At 30 docs	0.1127
At 100 docs	0.0826
At 200 docs	0.0672
At 500 docs	0.0457
At 1000 docs	0.0327

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1070



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	DCU97snt
Run Description	Category A, Automatic, short
Number of Topics	50

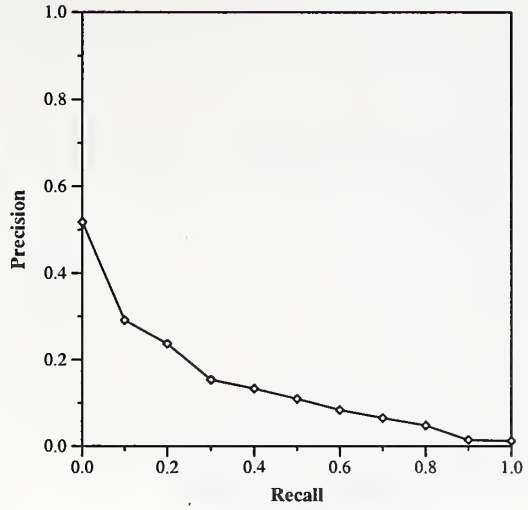
Total number of documents over all topics	
Retrieved:	50050
Relevant:	4611
Rel-ret:	1796

Recall Level Precision Averages	
Recall	Precision
0.00	0.5179
0.10	0.2917
0.20	0.2374
0.30	0.1542
0.40	0.1334
0.50	0.1099
0.60	0.0843
0.70	0.0655
0.80	0.0483
0.90	0.0144
1.00	0.0121

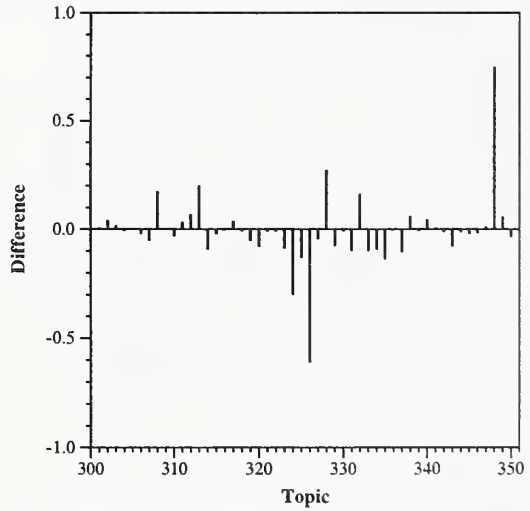
Average precision over all relevant docs	
non-interpolated	0.1296

Document Level Averages	
	Precision
At 5 docs	0.2800
At 10 docs	0.2000
At 15 docs	0.1840
At 20 docs	0.1700
At 30 docs	0.1500
At 100 docs	0.1104
At 200 docs	0.0813
At 500 docs	0.0519
At 1000 docs	0.0359

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1555



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	DCU97vs
Run Description	Category A, Automatic, title
Number of Topics	50

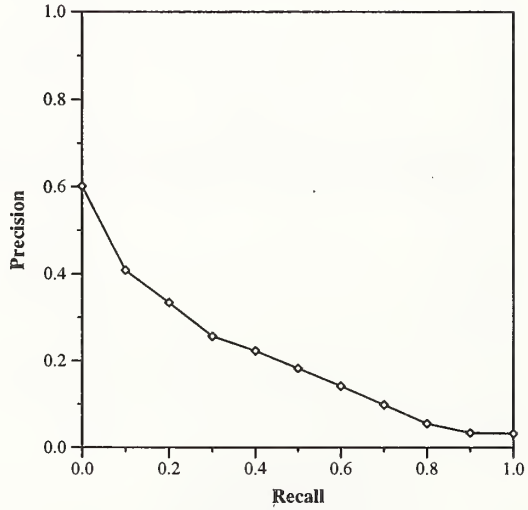
Total number of documents over all topics	
Retrieved:	50050
Relevant:	4611
Rel-ret:	2180

Recall Level Precision Averages	
Recall	Precision
0.00	0.6016
0.10	0.4086
0.20	0.3343
0.30	0.2570
0.40	0.2232
0.50	0.1830
0.60	0.1421
0.70	0.0988
0.80	0.0554
0.90	0.0341
1.00	0.0325

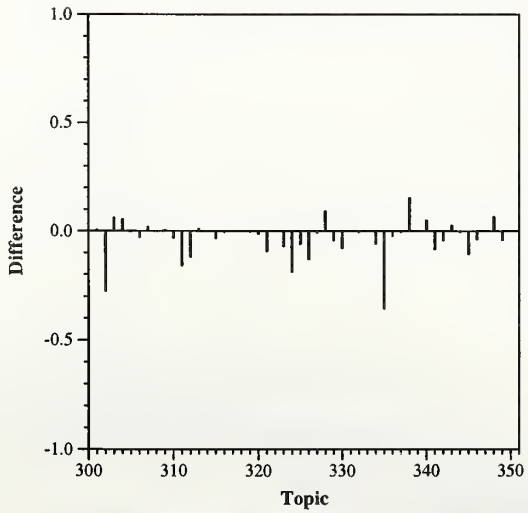
Average precision over all relevant docs	
non-interpolated	0.1941

Document Level Averages	
	Precision
At 5 docs	0.3800
At 10 docs	0.3280
At 15 docs	0.2973
At 20 docs	0.2840
At 30 docs	0.2553
At 100 docs	0.1644
At 200 docs	0.1166
At 500 docs	0.0702
At 1000 docs	0.0436

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2282



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	gerua3
Run Description	Category A, Automatic, short
Number of Topics	50

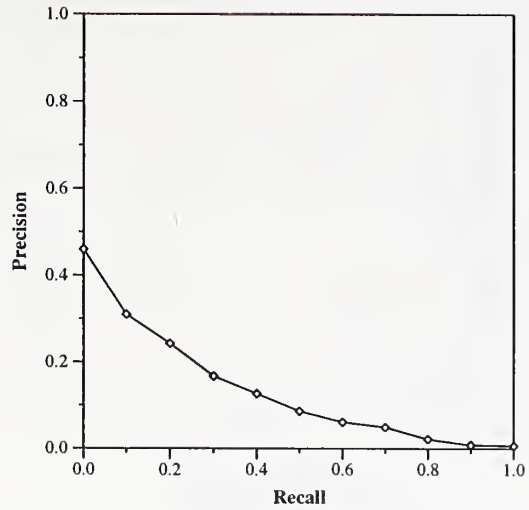
Total number of documents over all topics	
Retrieved:	49029
Relevant:	4611
Rel-ret:	1614

Recall Level Precision Averages	
Recall	Precision
0.00	0.4600
0.10	0.3098
0.20	0.2426
0.30	0.1671
0.40	0.1268
0.50	0.0865
0.60	0.0612
0.70	0.0496
0.80	0.0221
0.90	0.0084
1.00	0.0068

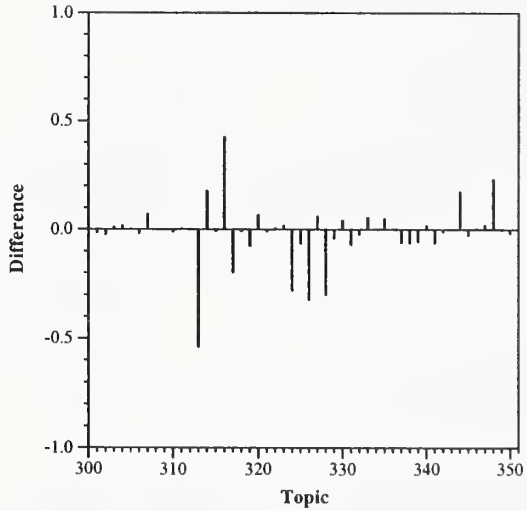
Average precision over all relevant docs	
non-interpolated	0.1201

Document Level Averages	
	Precision
At 5 docs	0.2640
At 10 docs	0.2500
At 15 docs	0.2400
At 20 docs	0.2270
At 30 docs	0.2013
At 100 docs	0.1186
At 200 docs	0.0828
At 500 docs	0.0480
At 1000 docs	0.0323

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1693



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	gmu97au1
Run Description	Category A, Automatic, short
Number of Topics	50

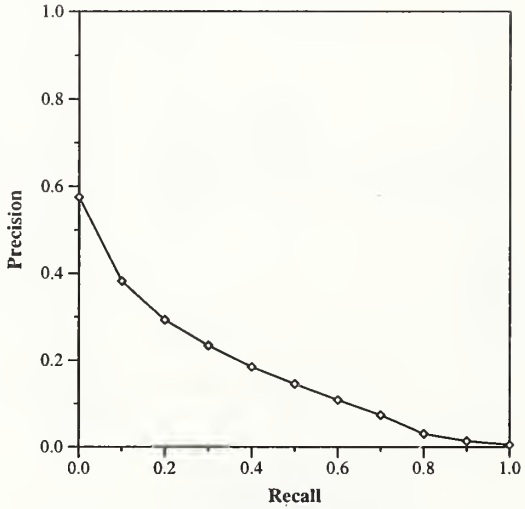
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2281

Recall Level Precision Averages	
Recall	Precision
0.00	0.5755
0.10	0.3825
0.20	0.2934
0.30	0.2342
0.40	0.1851
0.50	0.1457
0.60	0.1086
0.70	0.0740
0.80	0.0306
0.90	0.0139
1.00	0.0052

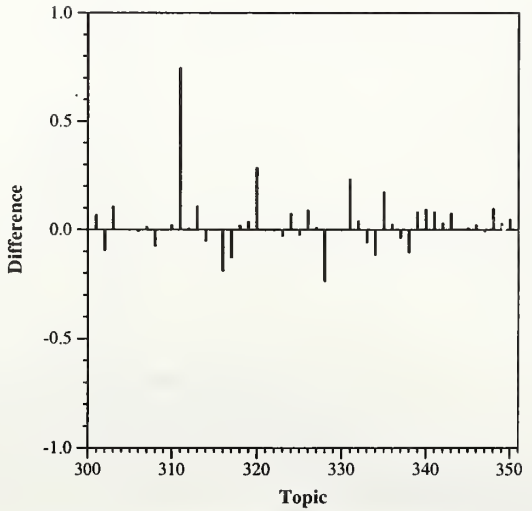
Average precision over all relevant docs	
non-interpolated	0.1661

Document Level Averages	
	Precision
At 5 docs	0.3880
At 10 docs	0.3520
At 15 docs	0.3173
At 20 docs	0.2970
At 30 docs	0.2700
At 100 docs	0.1780
At 200 docs	0.1230
At 500 docs	0.0708
At 1000 docs	0.0456

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2129



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	gmu97au2
Run Description	Category A, Automatic, short
Number of Topics	50

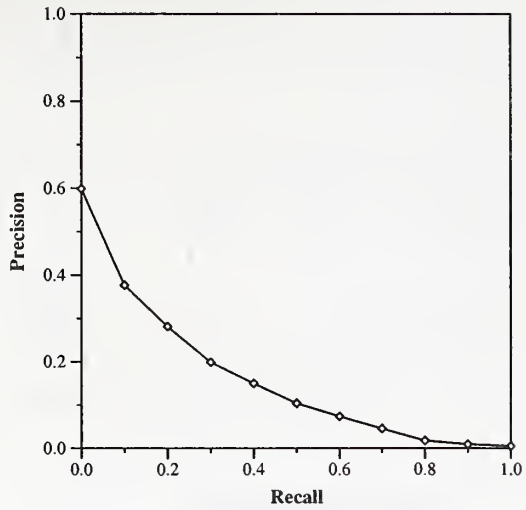
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2024

Recall Level Precision Averages	
Recall	Precision
0.00	0.5990
0.10	0.3771
0.20	0.2812
0.30	0.1989
0.40	0.1500
0.50	0.1040
0.60	0.0738
0.70	0.0456
0.80	0.0179
0.90	0.0093
1.00	0.0051

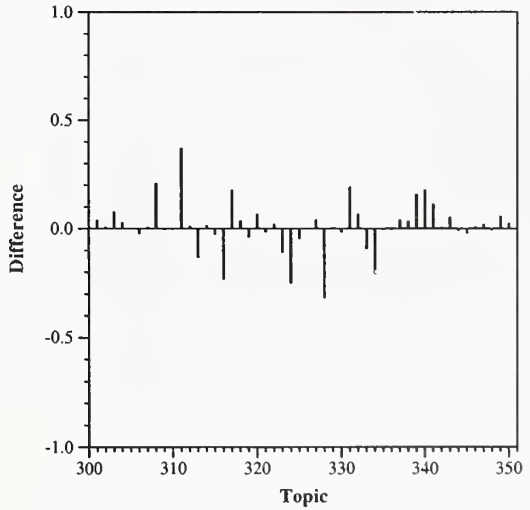
Average precision over all relevant docs	
non-interpolated	0.1469

Document Level Averages	
	Precision
At 5 docs	0.3720
At 10 docs	0.3220
At 15 docs	0.2947
At 20 docs	0.2730
At 30 docs	0.2440
At 100 docs	0.1532
At 200 docs	0.1058
At 500 docs	0.0618
At 1000 docs	0.0405

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1856



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ibmg97a
Run Description	Category A, Automatic, short
Number of Topics	50

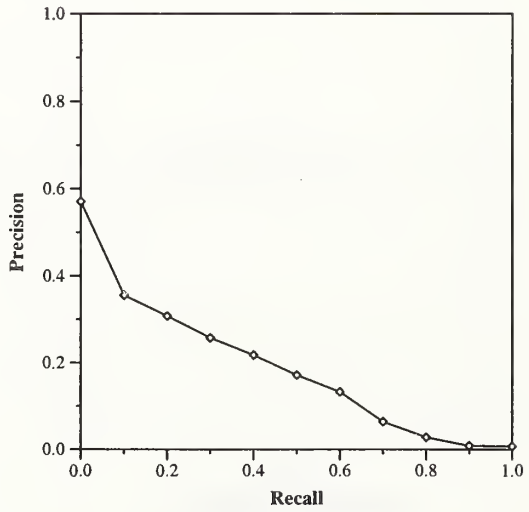
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1869

Recall Level Precision Averages	
Recall	Precision
0.00	0.5709
0.10	0.3557
0.20	0.3080
0.30	0.2578
0.40	0.2180
0.50	0.1716
0.60	0.1331
0.70	0.0642
0.80	0.0283
0.90	0.0086
1.00	0.0071

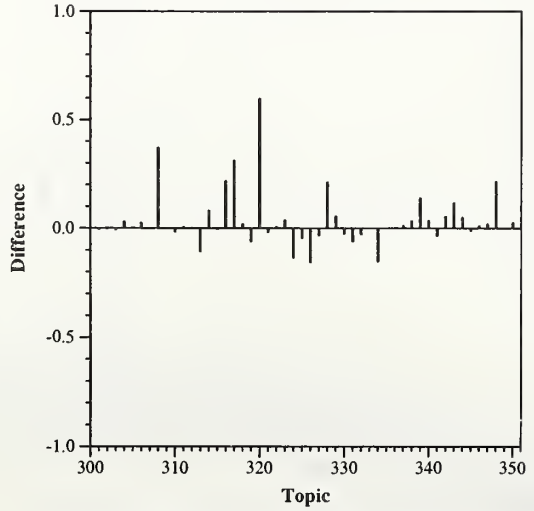
Average precision over all relevant docs	
non-interpolated	0.1727

Document Level Averages	
	Precision
At 5 docs	0.3480
At 10 docs	0.3140
At 15 docs	0.2813
At 20 docs	0.2620
At 30 docs	0.2340
At 100 docs	0.1422
At 200 docs	0.0976
At 500 docs	0.0588
At 1000 docs	0.0374

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2208



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ibmg97b
Run Description	Category A, Automatic, long
Number of Topics	50

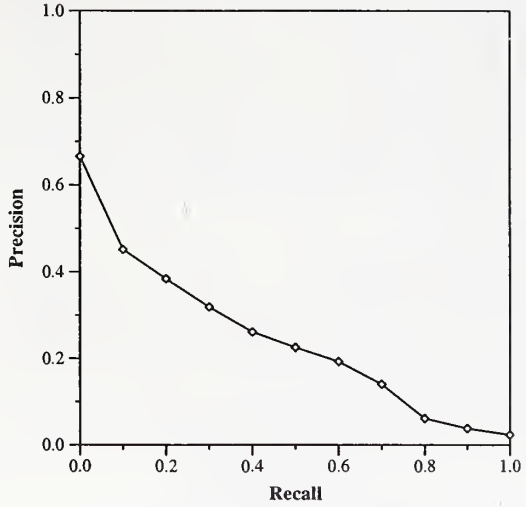
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2222

Recall Level Precision Averages	
Recall	Precision
0.00	0.6659
0.10	0.4516
0.20	0.3842
0.30	0.3191
0.40	0.2614
0.50	0.2255
0.60	0.1929
0.70	0.1404
0.80	0.0613
0.90	0.0383
1.00	0.0234

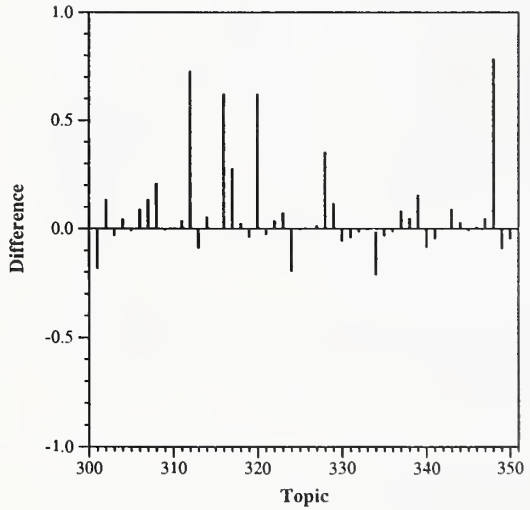
Average precision over all relevant docs	
non-interpolated	0.2309

Document Level Averages	
	Precision
At 5 docs	0.4560
At 10 docs	0.3920
At 15 docs	0.3507
At 20 docs	0.3240
At 30 docs	0.2920
At 100 docs	0.1766
At 200 docs	0.1216
At 500 docs	0.0715
At 1000 docs	0.0444

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2853



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ibms97a
Run Description	Category A, Automatic, short
Number of Topics	50

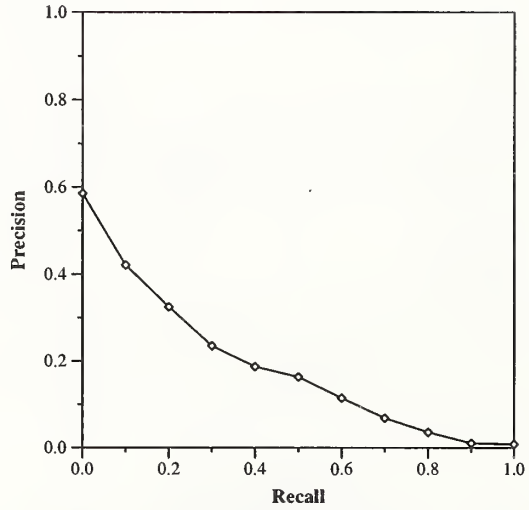
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2242

Recall Level Precision Averages	
Recall	Precision
0.00	0.5855
0.10	0.4212
0.20	0.3250
0.30	0.2351
0.40	0.1874
0.50	0.1634
0.60	0.1148
0.70	0.0688
0.80	0.0364
0.90	0.0108
1.00	0.0085

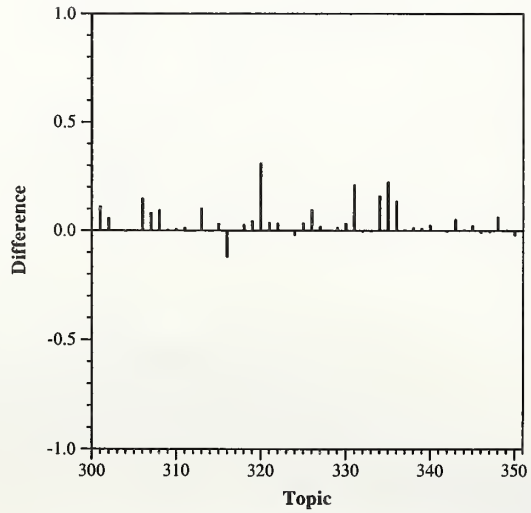
Average precision over all relevant docs	
non-interpolated	0.1775

Document Level Averages	
	Precision
At 5 docs	0.3880
At 10 docs	0.3660
At 15 docs	0.3240
At 20 docs	0.2930
At 30 docs	0.2667
At 100 docs	0.1734
At 200 docs	0.1200
At 500 docs	0.0695
At 1000 docs	0.0448

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2216



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Mercure1
Run Description	Category A, Automatic, long
Number of Topics	50

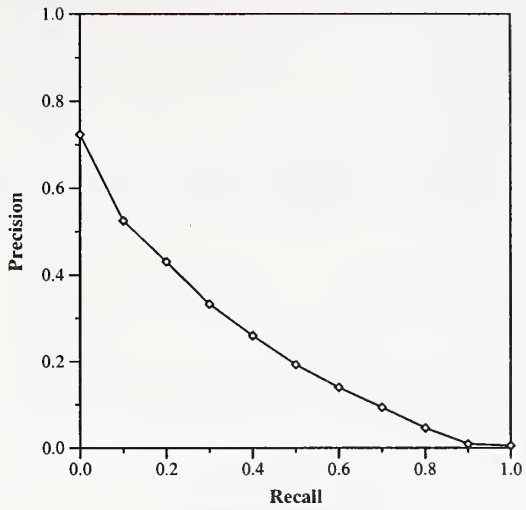
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2643

Recall Level Precision Averages	
Recall	Precision
0.00	0.7237
0.10	0.5253
0.20	0.4311
0.30	0.3331
0.40	0.2601
0.50	0.1932
0.60	0.1402
0.70	0.0940
0.80	0.0466
0.90	0.0096
1.00	0.0054

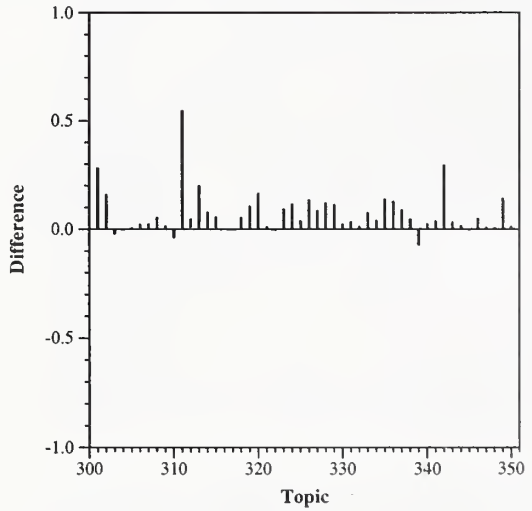
Average precision over all relevant docs	
non-interpolated	0.2305

Document Level Averages	
	Precision
At 5 docs	0.5360
At 10 docs	0.4640
At 15 docs	0.4080
At 20 docs	0.3700
At 30 docs	0.3260
At 100 docs	0.2110
At 200 docs	0.1494
At 500 docs	0.0849
At 1000 docs	0.0529

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2700



Recall-Precision Curve

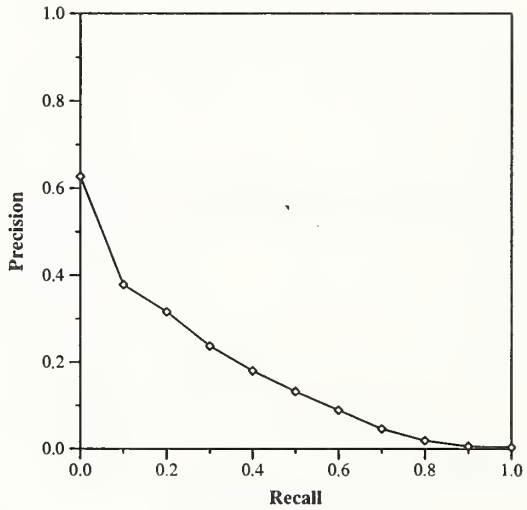


Difference from Median in Average Precision per Topic

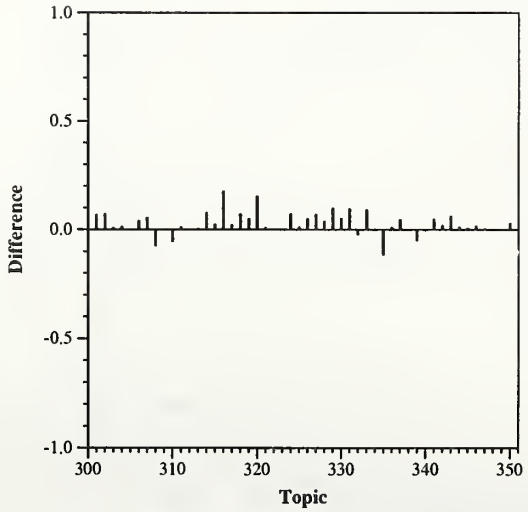
Summary Statistics	
Run Number	Mercure2
Run Description	Category A, Automatic, short
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2099

Recall Level Precision Averages	
Recall	Precision
0.00	0.6270
0.10	0.3787
0.20	0.3162
0.30	0.2377
0.40	0.1805
0.50	0.1327
0.60	0.0896
0.70	0.0465
0.80	0.0193
0.90	0.0062
1.00	0.0039
Average precision over all relevant docs	
non-interpolated	0.1640

Document Level Averages	
	Precision
At 5 docs	0.4440
At 10 docs	0.3520
At 15 docs	0.3173
At 20 docs	0.2950
At 30 docs	0.2593
At 100 docs	0.1642
At 200 docs	0.1131
At 500 docs	0.0656
At 1000 docs	0.0420
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2065



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Mercure3
Run Description	Category A, Automatic, title
Number of Topics	50

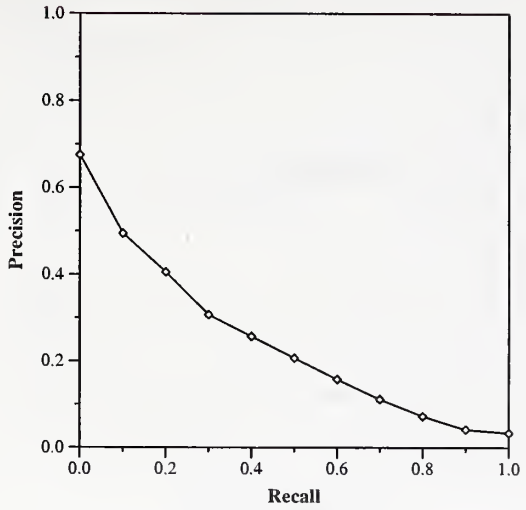
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2477

Recall Level Precision Averages	
Recall	Precision
0.00	0.6755
0.10	0.4947
0.20	0.4058
0.30	0.3073
0.40	0.2571
0.50	0.2071
0.60	0.1580
0.70	0.1119
0.80	0.0726
0.90	0.0419
1.00	0.0340

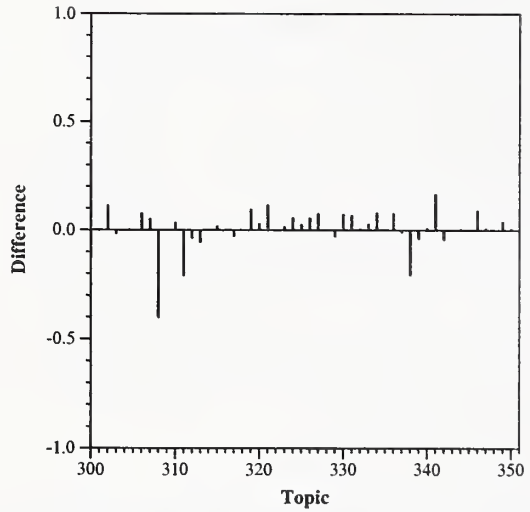
Average precision over all relevant docs	
non-interpolated	0.2316

Document Level Averages	
	Precision
At 5 docs	0.4560
At 10 docs	0.4240
At 15 docs	0.3800
At 20 docs	0.3570
At 30 docs	0.3267
At 100 docs	0.2010
At 200 docs	0.1396
At 500 docs	0.0806
At 1000 docs	0.0495

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2689



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	iss97s
Run Description	Category A, Automatic, short
Number of Topics	50

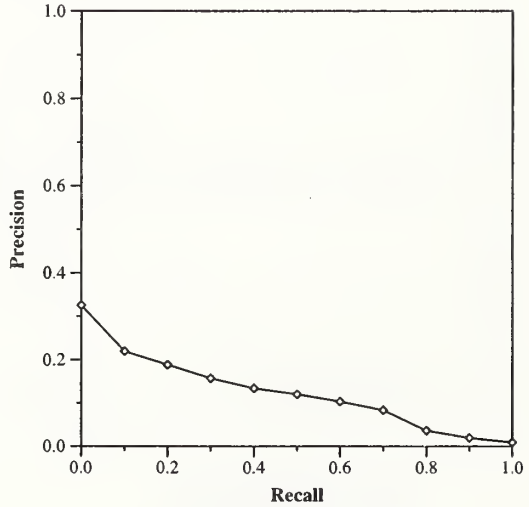
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1599

Recall Level Precision Averages	
Recall	Precision
0.00	0.3255
0.10	0.2196
0.20	0.1889
0.30	0.1577
0.40	0.1343
0.50	0.1206
0.60	0.1037
0.70	0.0838
0.80	0.0365
0.90	0.0196
1.00	0.0092

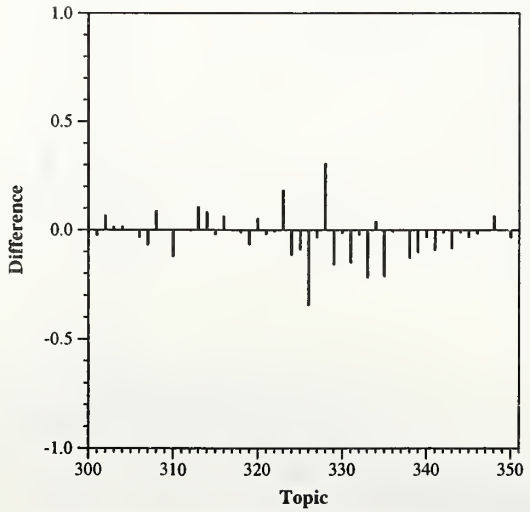
Average precision over all relevant docs	
non-interpolated	0.1135

Document Level Averages	
	Precision
At 5 docs	0.1600
At 10 docs	0.1500
At 15 docs	0.1387
At 20 docs	0.1380
At 30 docs	0.1300
At 100 docs	0.0944
At 200 docs	0.0657
At 500 docs	0.0438
At 1000 docs	0.0320

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1305



Recall-Precision Curve

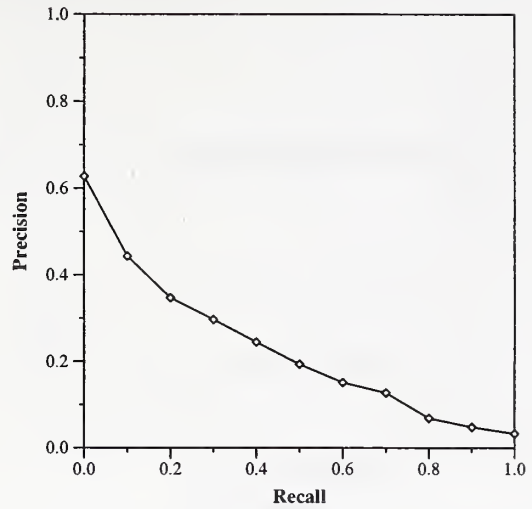


Difference from Median in Average Precision per Topic

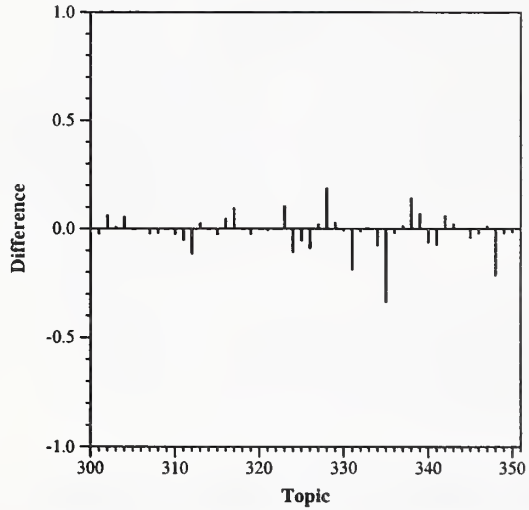
Summary Statistics	
Run Number	iss97vs
Run Description	Category A, Automatic, title
Number of Topics	50
Total number of documents over all topics	
Retrieved:	46038
Relevant:	4611
Rel-ret:	2238

Recall Level Precision Averages	
Recall	Precision
0.00	0.6276
0.10	0.4436
0.20	0.3476
0.30	0.2972
0.40	0.2450
0.50	0.1937
0.60	0.1517
0.70	0.1277
0.80	0.0690
0.90	0.0483
1.00	0.0331
Average precision over all relevant docs	
non-interpolated	0.2109

Document Level Averages	
	Precision
At 5 docs	0.3960
At 10 docs	0.3580
At 15 docs	0.3240
At 20 docs	0.3010
At 30 docs	0.2700
At 100 docs	0.1686
At 200 docs	0.1172
At 500 docs	0.0731
At 1000 docs	0.0448
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2454



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	LNaShort
Run Description	Category A, Automatic, short
Number of Topics	50

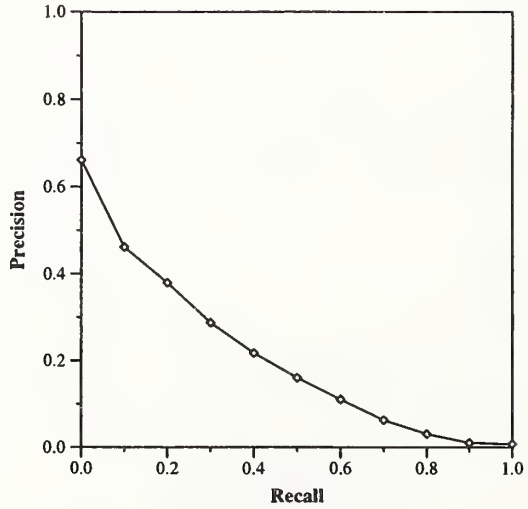
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2303

Recall Level Precision Averages	
Recall	Precision
0.00	0.6616
0.10	0.4616
0.20	0.3797
0.30	0.2877
0.40	0.2177
0.50	0.1604
0.60	0.1105
0.70	0.0626
0.80	0.0310
0.90	0.0102
1.00	0.0077

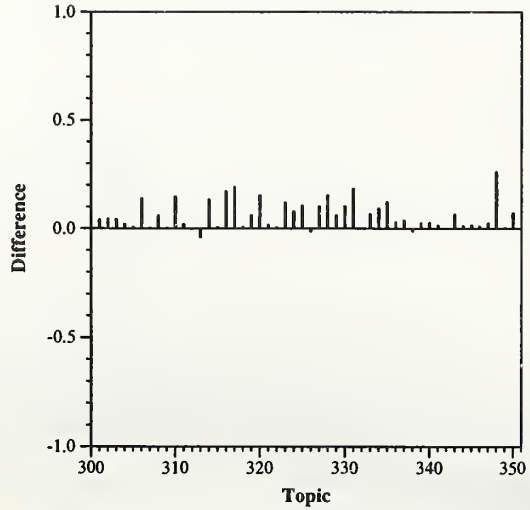
Average precision over all relevant docs	
non-interpolated	0.1972

Document Level Averages	
	Precision
At 5 docs	0.4600
At 10 docs	0.3960
At 15 docs	0.3613
At 20 docs	0.3210
At 30 docs	0.2800
At 100 docs	0.1708
At 200 docs	0.1197
At 500 docs	0.0703
At 1000 docs	0.0461

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2371



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	LNaVryShort
Run Description	Category A, Automatic, title
Number of Topics	50

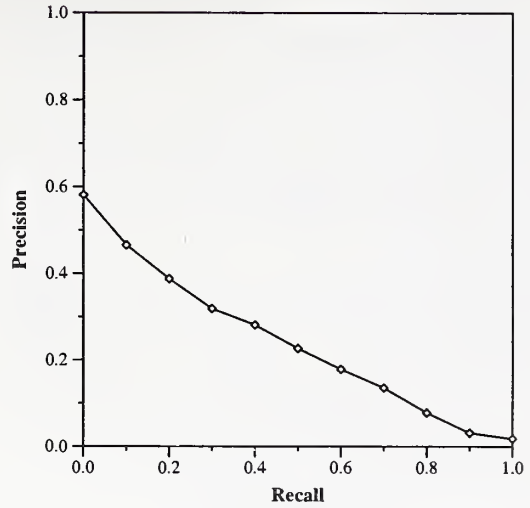
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2396

Recall Level Precision Averages	
Recall	Precision
0.00	0.5811
0.10	0.4659
0.20	0.3880
0.30	0.3190
0.40	0.2813
0.50	0.2271
0.60	0.1789
0.70	0.1361
0.80	0.0786
0.90	0.0317
1.00	0.0182

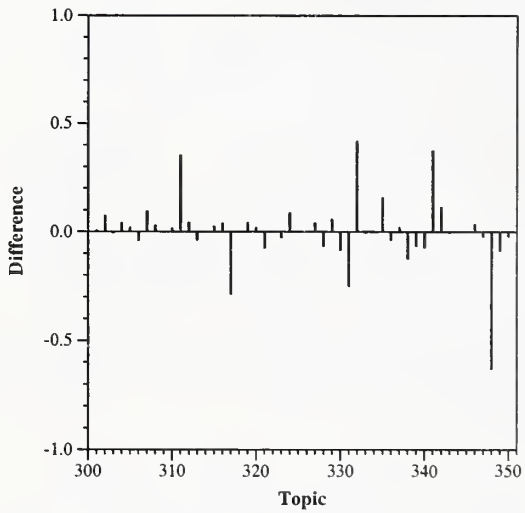
Average precision over all relevant docs	
non-interpolated	0.2283

Document Level Averages	
	Precision
At 5 docs	0.4200
At 10 docs	0.3840
At 15 docs	0.3507
At 20 docs	0.3310
At 30 docs	0.3040
At 100 docs	0.2044
At 200 docs	0.1396
At 500 docs	0.0774
At 1000 docs	0.0479

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2648



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	mds601
Run Description	Category A, Automatic, short
Number of Topics	50

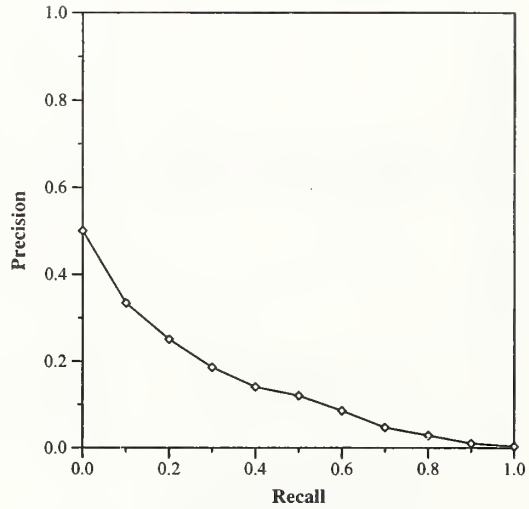
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1776

Recall Level Precision Averages	
Recall	Precision
0.00	0.5006
0.10	0.3343
0.20	0.2511
0.30	0.1865
0.40	0.1411
0.50	0.1209
0.60	0.0859
0.70	0.0471
0.80	0.0290
0.90	0.0104
1.00	0.0038

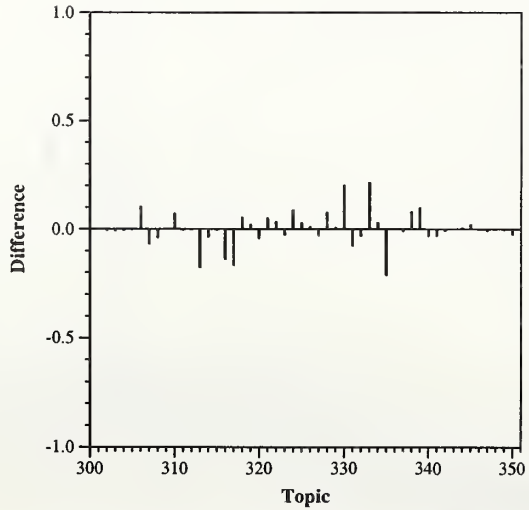
Average precision over all relevant docs	
non-interpolated	0.1375

Document Level Averages	
	Precision
At 5 docs	0.3480
At 10 docs	0.2940
At 15 docs	0.2600
At 20 docs	0.2440
At 30 docs	0.2213
At 100 docs	0.1298
At 200 docs	0.0926
At 500 docs	0.0539
At 1000 docs	0.0355

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1700



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	mds602
Run Description	Category A, Automatic, long
Number of Topics	50

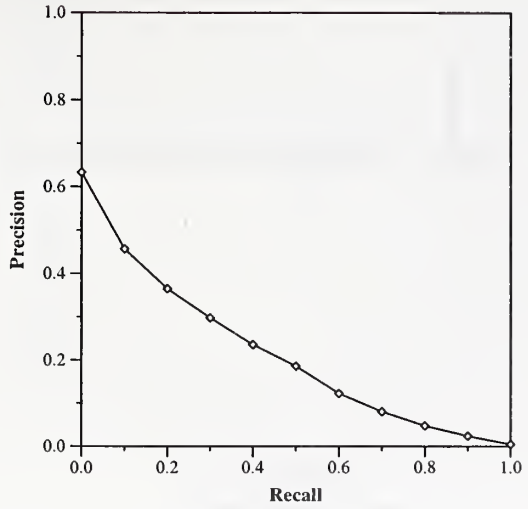
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2471

Recall Level Precision Averages	
Recall	Precision
0.00	0.6333
0.10	0.4568
0.20	0.3648
0.30	0.2978
0.40	0.2358
0.50	0.1861
0.60	0.1229
0.70	0.0807
0.80	0.0480
0.90	0.0241
1.00	0.0049

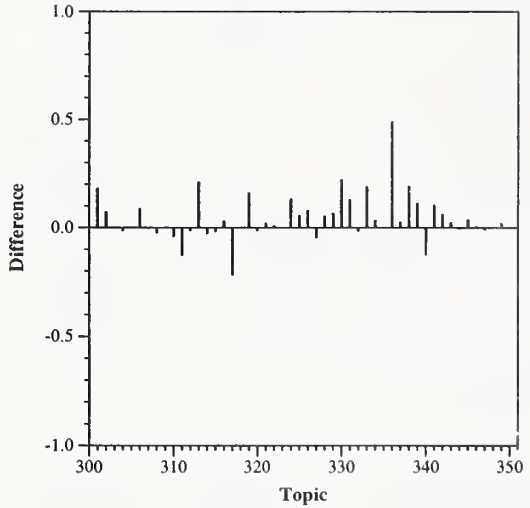
Average precision over all relevant docs	
non-interpolated	0.2017

Document Level Averages	
	Precision
At 5 docs	0.4160
At 10 docs	0.3740
At 15 docs	0.3467
At 20 docs	0.3240
At 30 docs	0.2880
At 100 docs	0.1814
At 200 docs	0.1292
At 500 docs	0.0779
At 1000 docs	0.0494

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2303



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	mds603
Run Description	Category A, Automatic, title
Number of Topics	50

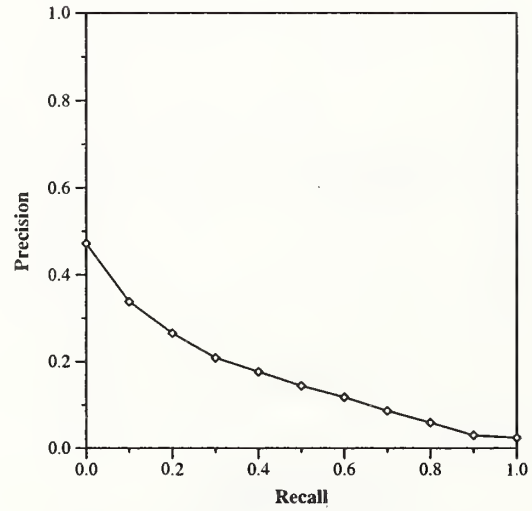
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1919

Recall Level Precision Averages	
Recall	Precision
0.00	0.4723
0.10	0.3385
0.20	0.2656
0.30	0.2089
0.40	0.1769
0.50	0.1444
0.60	0.1179
0.70	0.0864
0.80	0.0593
0.90	0.0299
1.00	0.0241

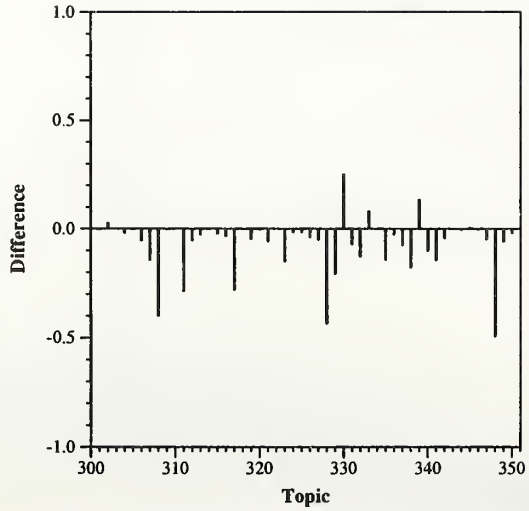
Average precision over all relevant docs	
non-interpolated	0.1574

Document Level Averages	
	Precision
At 5 docs	0.3080
At 10 docs	0.2880
At 15 docs	0.2800
At 20 docs	0.2610
At 30 docs	0.2380
At 100 docs	0.1446
At 200 docs	0.0981
At 500 docs	0.0556
At 1000 docs	0.0384

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1877



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	jalbse
Run Description	Category A, Automatic, short
Number of Topics	50

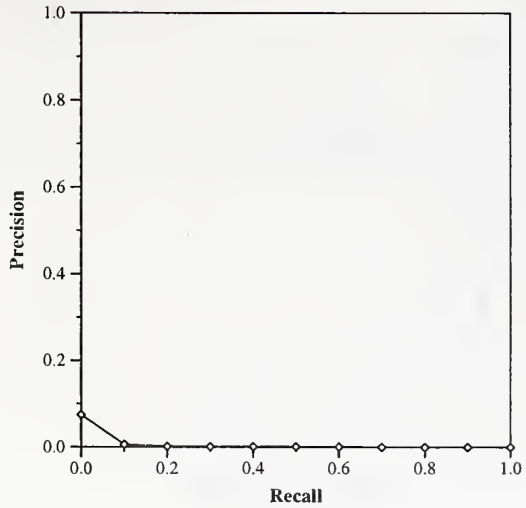
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	221

Recall Level Precision Averages	
Recall	Precision
0.00	0.0747
0.10	0.0061
0.20	0.0020
0.30	0.0015
0.40	0.0015
0.50	0.0011
0.60	0.0009
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000

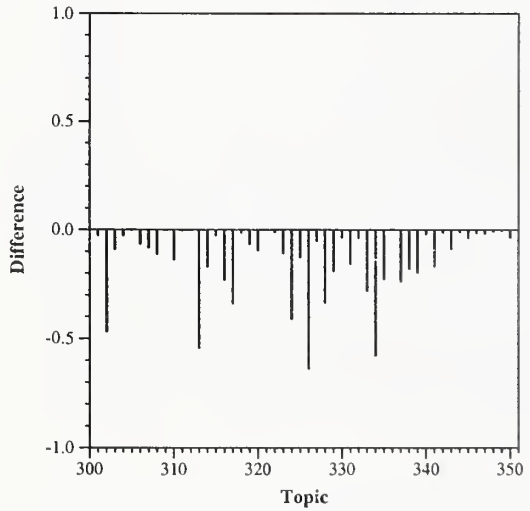
Average precision over all relevant docs	
non-interpolated	0.0026

Document Level Averages	
	Precision
At 5 docs	0.0280
At 10 docs	0.0240
At 15 docs	0.0213
At 20 docs	0.0190
At 30 docs	0.0147
At 100 docs	0.0106
At 200 docs	0.0075
At 500 docs	0.0056
At 1000 docs	0.0044

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0082



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	jalbse0
Run Description	Category A, Automatic, short
Number of Topics	50

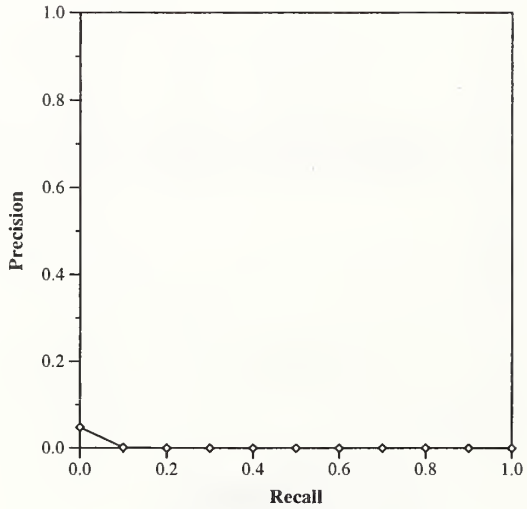
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	91

Recall Level Precision Averages	
Recall	Precision
0.00	0.0478
0.10	0.0012
0.20	0.0000
0.30	0.0000
0.40	0.0000
0.50	0.0000
0.60	0.0000
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000

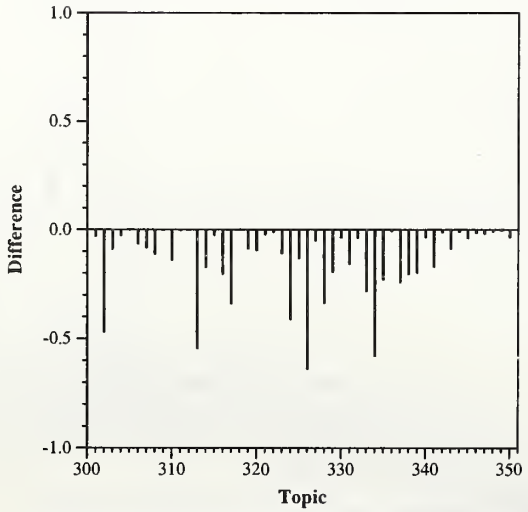
Average precision over all relevant docs	
non-interpolated	0.0012

Document Level Averages	
	Precision
At 5 docs	0.0160
At 10 docs	0.0120
At 15 docs	0.0133
At 20 docs	0.0110
At 30 docs	0.0080
At 100 docs	0.0064
At 200 docs	0.0045
At 500 docs	0.0026
At 1000 docs	0.0018

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0050



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	nmsu1
Run Description	Category A, Automatic, long
Number of Topics	50

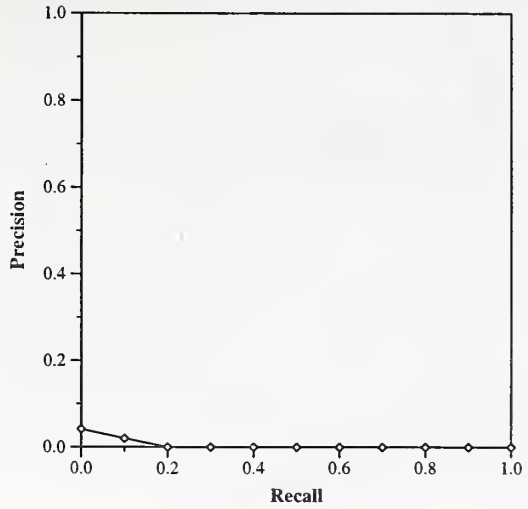
Total number of documents over all topics	
Retrieved:	5034
Relevant:	4611
Rel-ret:	52

Recall Level Precision Averages	
Recall	Precision
0.00	0.0412
0.10	0.0200
0.20	0.0000
0.30	0.0000
0.40	0.0000
0.50	0.0000
0.60	0.0000
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000

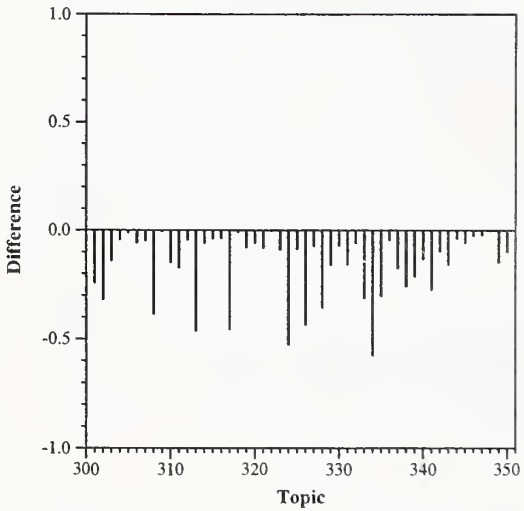
Average precision over all relevant docs	
non-interpolated	0.0028

Document Level Averages	
	Precision
At 5 docs	0.0280
At 10 docs	0.0180
At 15 docs	0.0187
At 20 docs	0.0160
At 30 docs	0.0127
At 100 docs	0.0050
At 200 docs	0.0028
At 500 docs	0.0018
At 1000 docs	0.0010

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0054



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	nsasg1
Run Description	Category A, Automatic, long
Number of Topics	50

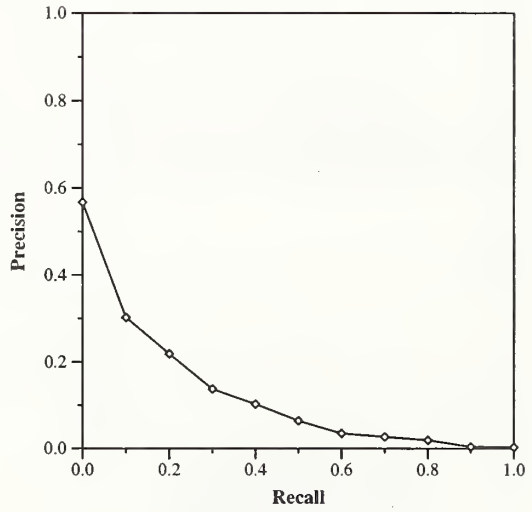
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1656

Recall Level Precision Averages	
Recall	Precision
0.00	0.5672
0.10	0.3024
0.20	0.2188
0.30	0.1374
0.40	0.1028
0.50	0.0643
0.60	0.0351
0.70	0.0273
0.80	0.0195
0.90	0.0044
1.00	0.0035

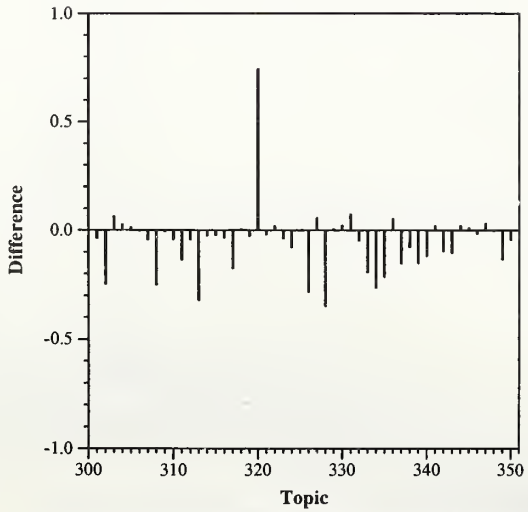
Average precision over all relevant docs	
non-interpolated	0.1071

Document Level Averages	
	Precision
At 5 docs	0.3280
At 10 docs	0.2720
At 15 docs	0.2507
At 20 docs	0.2230
At 30 docs	0.1900
At 100 docs	0.1264
At 200 docs	0.0854
At 500 docs	0.0506
At 1000 docs	0.0331

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1656



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	nsasg2
Run Description	Category A, Automatic, short
Number of Topics	50

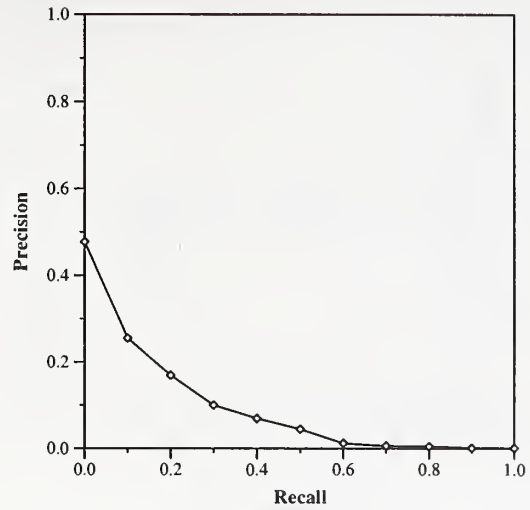
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1014

Recall Level Precision Averages	
Recall	Precision
0.00	0.4775
0.10	0.2553
0.20	0.1694
0.30	0.1005
0.40	0.0697
0.50	0.0449
0.60	0.0121
0.70	0.0062
0.80	0.0052
0.90	0.0014
1.00	0.0014

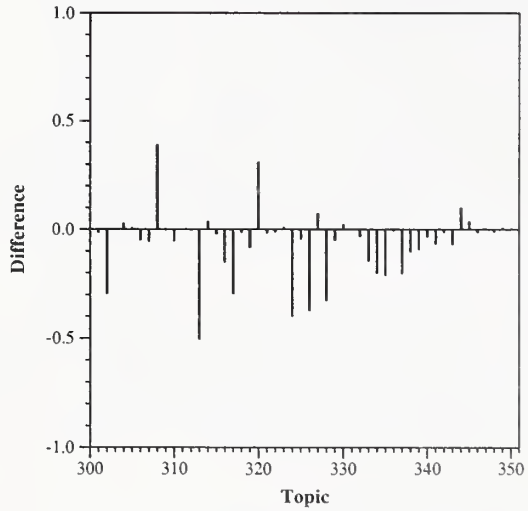
Average precision over all relevant docs	
non-interpolated	0.0787

Document Level Averages	
	Precision
At 5 docs	0.2720
At 10 docs	0.2320
At 15 docs	0.1893
At 20 docs	0.1700
At 30 docs	0.1393
At 100 docs	0.0768
At 200 docs	0.0543
At 500 docs	0.0319
At 1000 docs	0.0203

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1249



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	pirc7Aa
Run Description	Category A, Automatic, long
Number of Topics	50

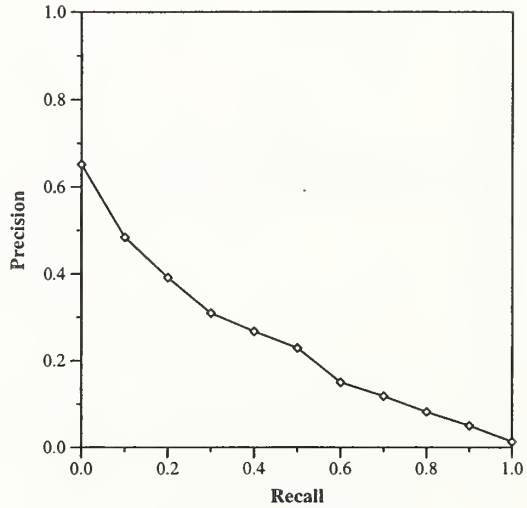
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2674

Recall Level Precision Averages	
Recall	Precision
0.00	0.6518
0.10	0.4844
0.20	0.3917
0.30	0.3099
0.40	0.2681
0.50	0.2297
0.60	0.1505
0.70	0.1186
0.80	0.0820
0.90	0.0504
1.00	0.0133

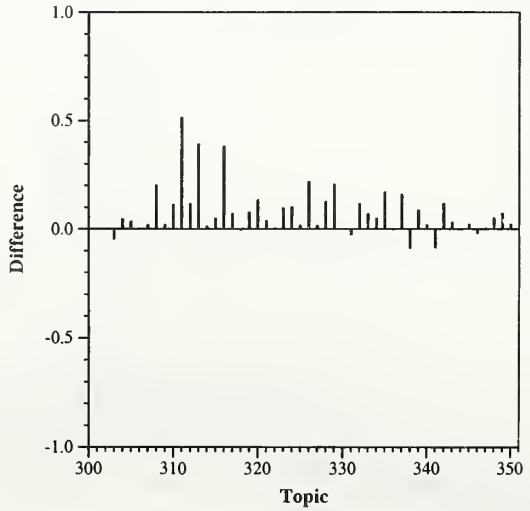
Average precision over all relevant docs	
non-interpolated	0.2332

Document Level Averages	
	Precision
At 5 docs	0.4920
At 10 docs	0.4260
At 15 docs	0.3773
At 20 docs	0.3460
At 30 docs	0.3093
At 100 docs	0.2120
At 200 docs	0.1490
At 500 docs	0.0849
At 1000 docs	0.0535

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2722



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	pirc7Ad
Run Description	Category A, Automatic, short
Number of Topics	50

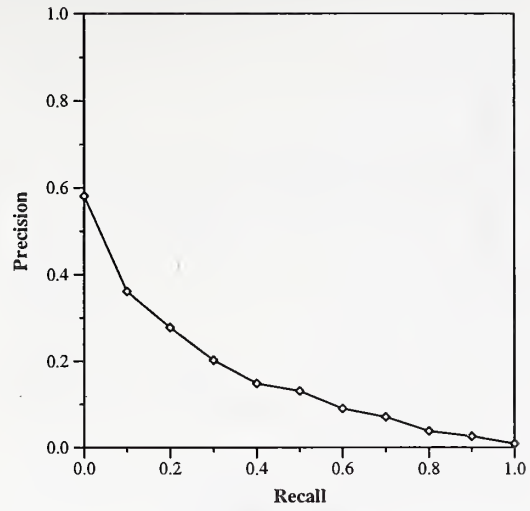
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1718

Recall Level Precision Averages	
Recall	Precision
0.00	0.5812
0.10	0.3615
0.20	0.2779
0.30	0.2025
0.40	0.1485
0.50	0.1309
0.60	0.0902
0.70	0.0706
0.80	0.0380
0.90	0.0260
1.00	0.0090

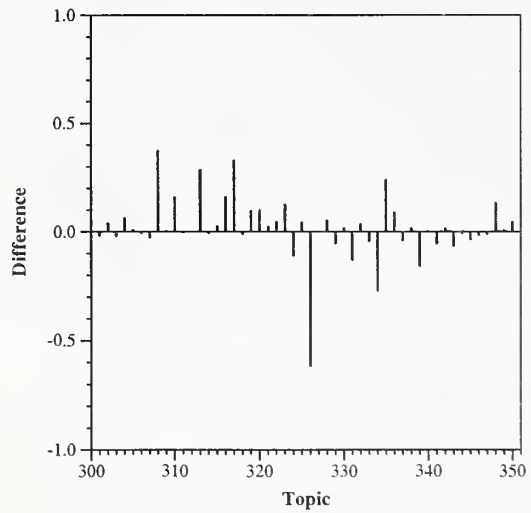
Average precision over all relevant docs	
non-interpolated	0.1533

Document Level Averages	
	Precision
At 5 docs	0.3440
At 10 docs	0.2940
At 15 docs	0.2640
At 20 docs	0.2450
At 30 docs	0.2133
At 100 docs	0.1330
At 200 docs	0.0908
At 500 docs	0.0533
At 1000 docs	0.0344

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1974



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	pirc7At
Run Description	Category A, Automatic, title
Number of Topics	50

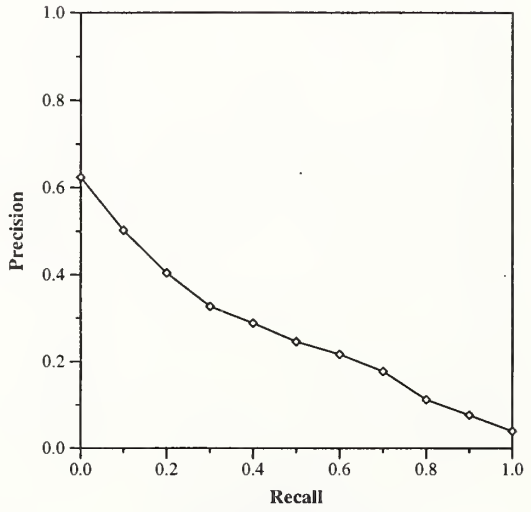
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2377

Recall Level Precision Averages	
Recall	Precision
0.00	0.6239
0.10	0.5024
0.20	0.4041
0.30	0.3270
0.40	0.2888
0.50	0.2461
0.60	0.2170
0.70	0.1777
0.80	0.1123
0.90	0.0767
1.00	0.0396

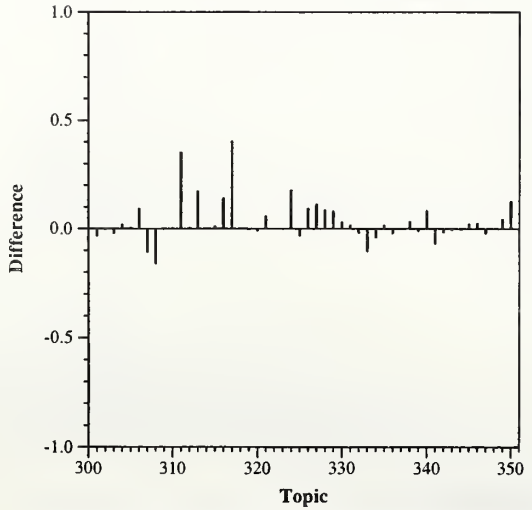
Average precision over all relevant docs	
non-interpolated	0.2556

Document Level Averages	
	Precision
At 5 docs	0.4320
At 10 docs	0.4020
At 15 docs	0.3707
At 20 docs	0.3390
At 30 docs	0.3093
At 100 docs	0.2092
At 200 docs	0.1499
At 500 docs	0.0796
At 1000 docs	0.0475

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2912



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Brkly21
Run Description	Category A, Automatic, short
Number of Topics	50

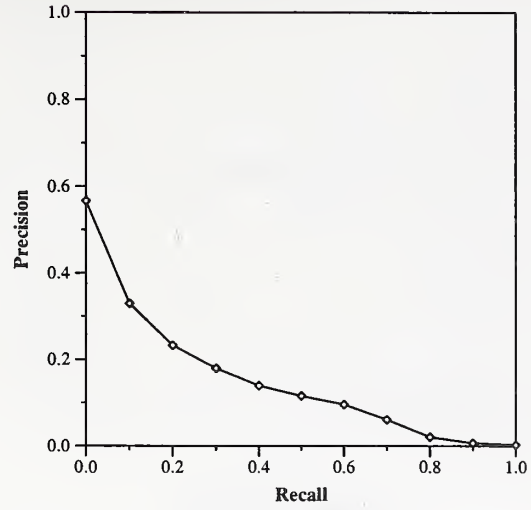
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1615

Recall Level Precision Averages	
Recall	Precision
0.00	0.5668
0.10	0.3298
0.20	0.2333
0.30	0.1802
0.40	0.1399
0.50	0.1160
0.60	0.0963
0.70	0.0611
0.80	0.0209
0.90	0.0070
1.00	0.0030

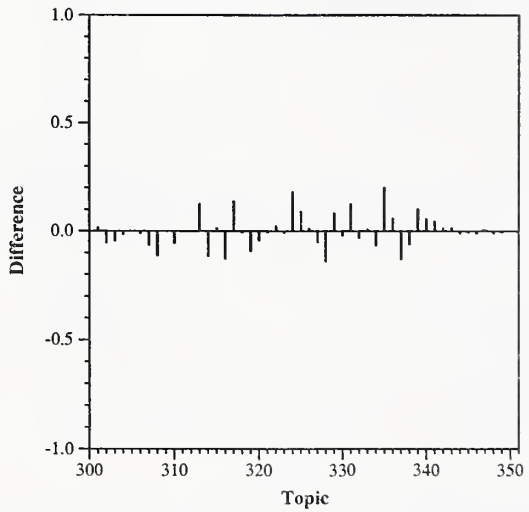
Average precision over all relevant docs	
non-interpolated	0.1376

Document Level Averages	
	Precision
At 5 docs	0.3680
At 10 docs	0.2940
At 15 docs	0.2680
At 20 docs	0.2450
At 30 docs	0.2160
At 100 docs	0.1286
At 200 docs	0.0878
At 500 docs	0.0502
At 1000 docs	0.0323

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1675



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Brkly22
Run Description	Category A, Automatic, long
Number of Topics	50

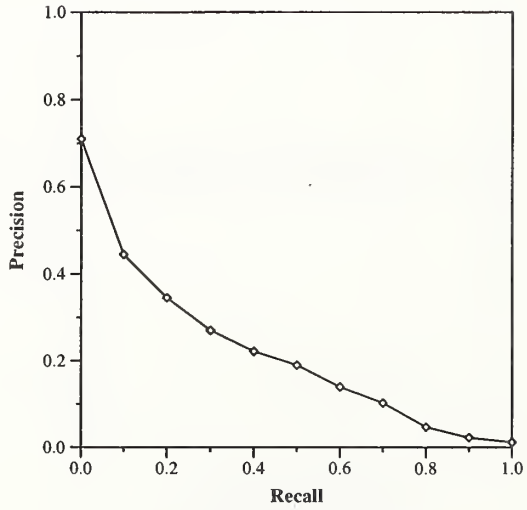
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2547

Recall Level Precision Averages	
Recall	Precision
0.00	0.7105
0.10	0.4449
0.20	0.3456
0.30	0.2704
0.40	0.2218
0.50	0.1903
0.60	0.1398
0.70	0.1029
0.80	0.0472
0.90	0.0226
1.00	0.0119

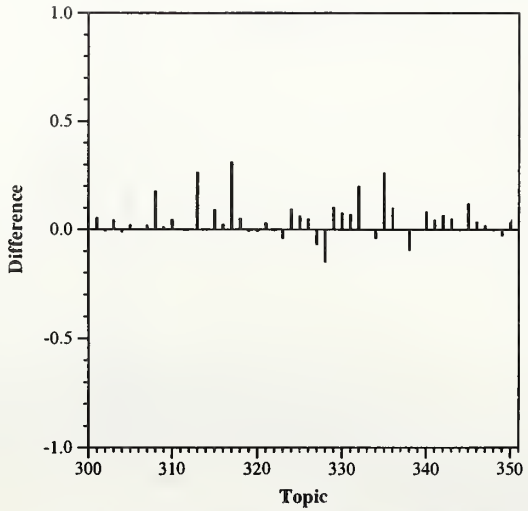
Average precision over all relevant docs	
non-interpolated	0.2021

Document Level Averages	
	Precision
At 5 docs	0.4680
At 10 docs	0.4080
At 15 docs	0.3693
At 20 docs	0.3440
At 30 docs	0.3053
At 100 docs	0.1932
At 200 docs	0.1365
At 500 docs	0.0800
At 1000 docs	0.0509

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2422



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	glair61
Run Description	Category A, Automatic, title
Number of Topics	50

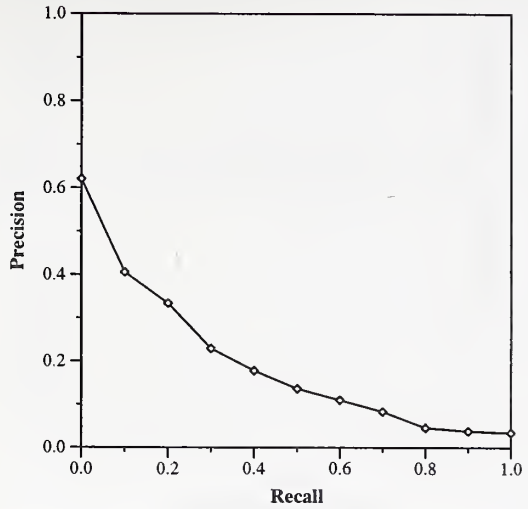
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1964

Recall Level Precision Averages	
Recall	Precision
0.00	0.6208
0.10	0.4059
0.20	0.3337
0.30	0.2296
0.40	0.1781
0.50	0.1364
0.60	0.1101
0.70	0.0835
0.80	0.0460
0.90	0.0383
1.00	0.0345

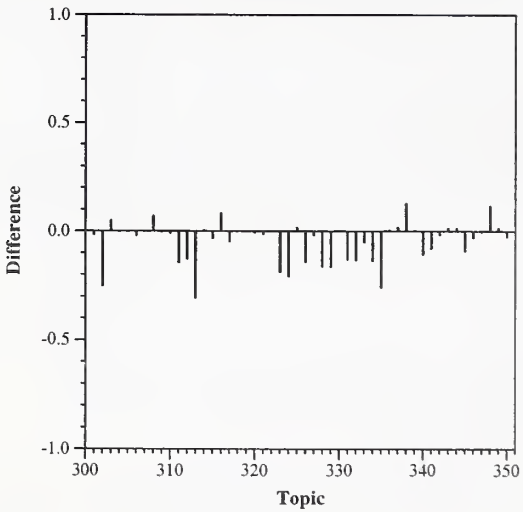
Average precision over all relevant docs	
non-interpolated	0.1772

Document Level Averages	
	Precision
At 5 docs	0.3760
At 10 docs	0.3060
At 15 docs	0.2867
At 20 docs	0.2620
At 30 docs	0.2387
At 100 docs	0.1394
At 200 docs	0.0972
At 500 docs	0.0614
At 1000 docs	0.0393

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2094



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	glair64
Run Description	Category A, Automatic, short
Number of Topics	50

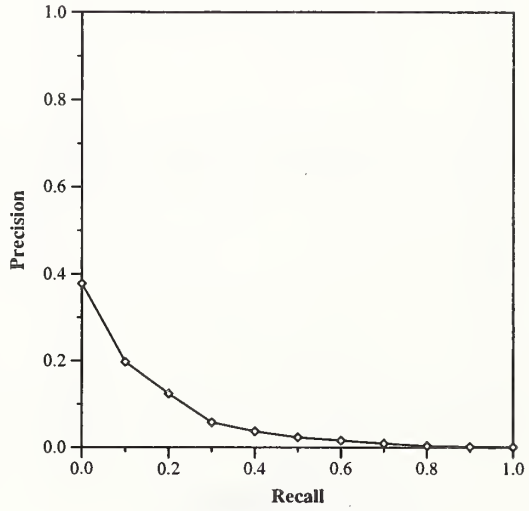
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1239

Recall Level Precision Averages	
Recall	Precision
0.00	0.3781
0.10	0.1976
0.20	0.1245
0.30	0.0580
0.40	0.0372
0.50	0.0233
0.60	0.0163
0.70	0.0097
0.80	0.0039
0.90	0.0022
1.00	0.0013

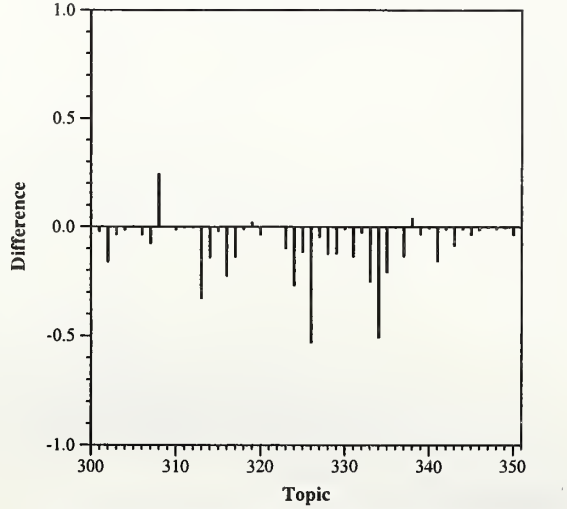
Average precision over all relevant docs	
non-interpolated	0.0585

Document Level Averages	
	Precision
At 5 docs	0.1600
At 10 docs	0.1240
At 15 docs	0.1120
At 20 docs	0.1040
At 30 docs	0.0933
At 100 docs	0.0606
At 200 docs	0.0469
At 500 docs	0.0323
At 1000 docs	0.0248

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0814



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	umcpa197
Run Description	Category A, Automatic, short
Number of Topics	50

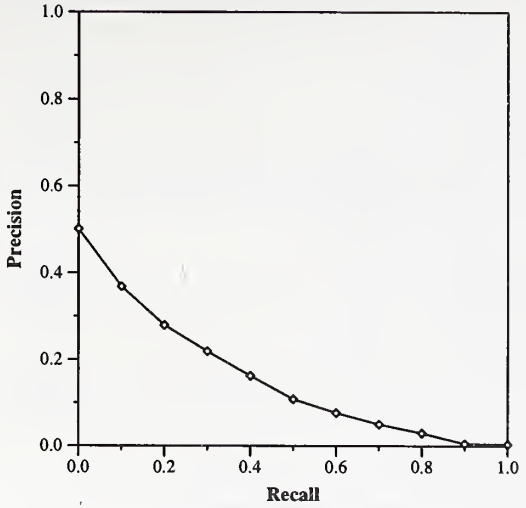
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1894

Recall Level Precision Averages	
Recall	Precision
0.00	0.5011
0.10	0.3686
0.20	0.2792
0.30	0.2189
0.40	0.1624
0.50	0.1082
0.60	0.0765
0.70	0.0504
0.80	0.0298
0.90	0.0052
1.00	0.0038

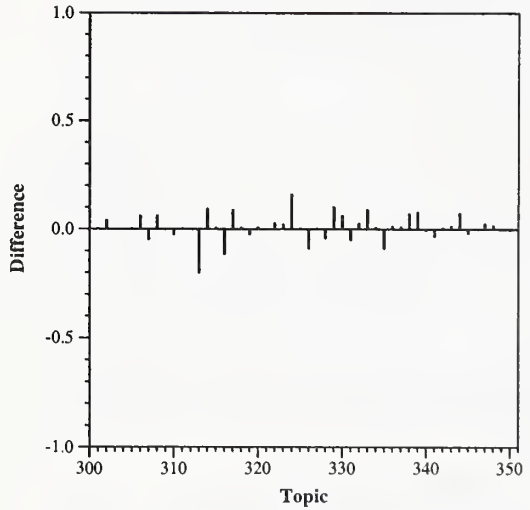
Average precision over all relevant docs	
non-interpolated	0.1460

Document Level Averages	
	Precision
At 5 docs	0.3360
At 10 docs	0.3080
At 15 docs	0.2893
At 20 docs	0.2570
At 30 docs	0.2447
At 100 docs	0.1452
At 200 docs	0.1007
At 500 docs	0.0585
At 1000 docs	0.0379

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1925



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	INQ401
Run Description	Category A, Automatic, short
Number of Topics	50

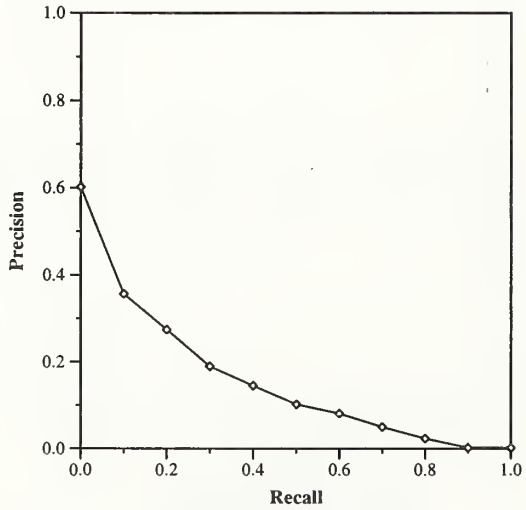
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1375

Recall Level Precision Averages	
Recall	Precision
0.00	0.6023
0.10	0.3567
0.20	0.2746
0.30	0.1899
0.40	0.1454
0.50	0.1024
0.60	0.0812
0.70	0.0502
0.80	0.0235
0.90	0.0023
1.00	0.0023

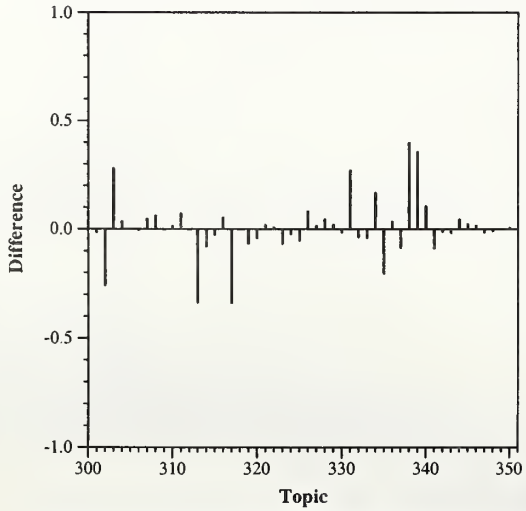
Average precision over all relevant docs	
non-interpolated	0.1440

Document Level Averages	
	Precision
At 5 docs	0.4200
At 10 docs	0.3340
At 15 docs	0.2880
At 20 docs	0.2630
At 30 docs	0.2247
At 100 docs	0.1296
At 200 docs	0.0827
At 500 docs	0.0452
At 1000 docs	0.0275

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1835



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	INQ402
Run Description	Category A, Automatic, long
Number of Topics	50

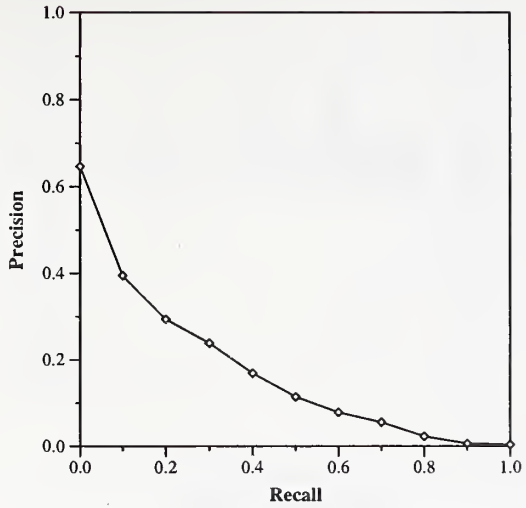
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1512

Recall Level Precision Averages	
Recall	Precision
0.00	0.6468
0.10	0.3951
0.20	0.2941
0.30	0.2388
0.40	0.1692
0.50	0.1148
0.60	0.0784
0.70	0.0557
0.80	0.0228
0.90	0.0061
1.00	0.0037

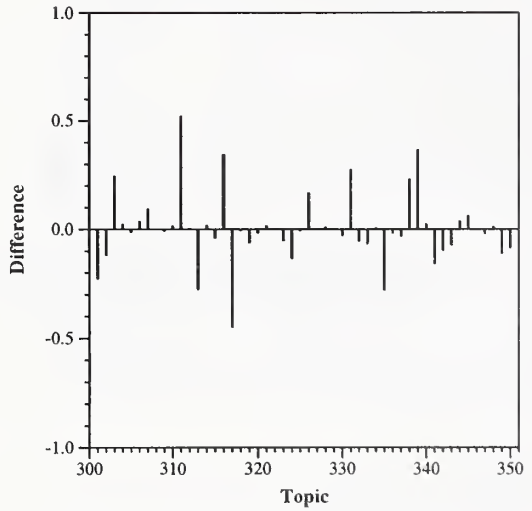
Average precision over all relevant docs	
non-interpolated	0.1612

Document Level Averages	
	Precision
At 5 docs	0.4560
At 10 docs	0.3740
At 15 docs	0.3160
At 20 docs	0.2880
At 30 docs	0.2513
At 100 docs	0.1470
At 200 docs	0.0957
At 500 docs	0.0513
At 1000 docs	0.0302

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1937



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ispa1
Run Description	Category A, Automatic, long
Number of Topics	50

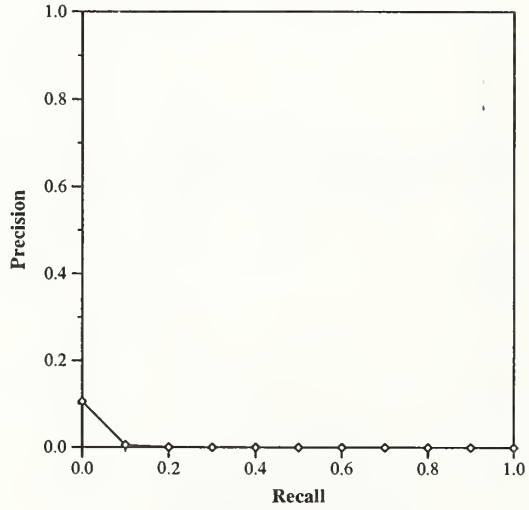
Total number of documents over all topics	
Retrieved:	49999
Relevant:	4611
Rel-ret:	277

Recall Level Precision Averages	
Recall	Precision
0.00	0.1056
0.10	0.0054
0.20	0.0005
0.30	0.0000
0.40	0.0000
0.50	0.0000
0.60	0.0000
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000

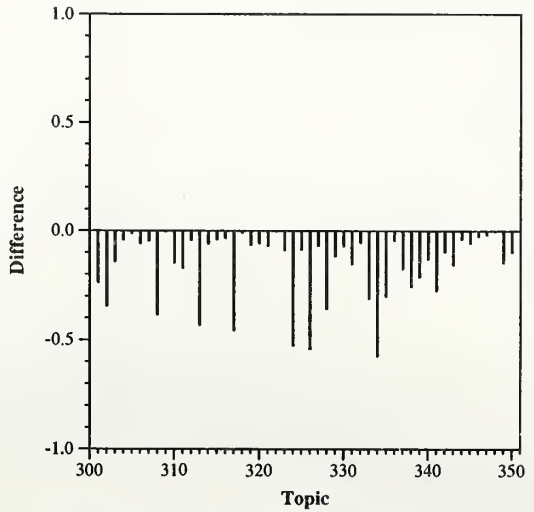
Average precision over all relevant docs	
non-interpolated	0.0030

Document Level Averages	
	Precision
At 5 docs	0.0360
At 10 docs	0.0260
At 15 docs	0.0240
At 20 docs	0.0240
At 30 docs	0.0240
At 100 docs	0.0164
At 200 docs	0.0120
At 500 docs	0.0072
At 1000 docs	0.0055

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0127



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ispa2
Run Description	Category A, Automatic, short
Number of Topics	50

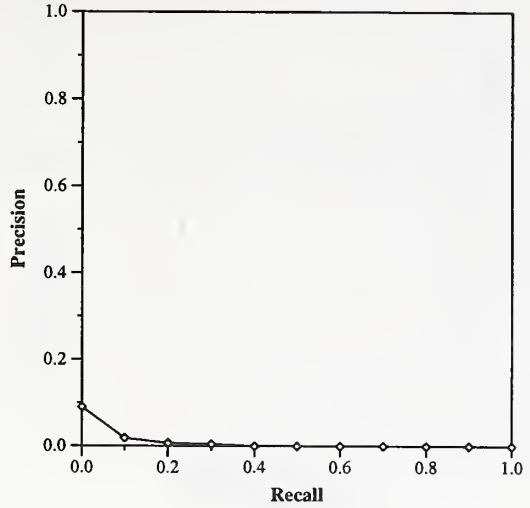
Total number of documents over all topics	
Retrieved:	43528
Relevant:	4611
Rel-ret:	496

Recall Level Precision Averages	
Recall	Precision
0.00	0.0899
0.10	0.0182
0.20	0.0068
0.30	0.0044
0.40	0.0000
0.50	0.0000
0.60	0.0000
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000

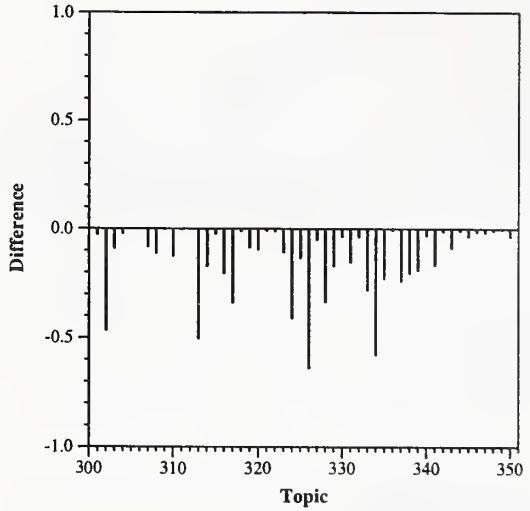
Average precision over all relevant docs	
non-interpolated	0.0050

Document Level Averages	
	Precision
At 5 docs	0.0200
At 10 docs	0.0140
At 15 docs	0.0173
At 20 docs	0.0200
At 30 docs	0.0180
At 100 docs	0.0214
At 200 docs	0.0184
At 500 docs	0.0136
At 1000 docs	0.0099

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0202



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	uwmt6a1
Run Description	Category A, Automatic, title
Number of Topics	50

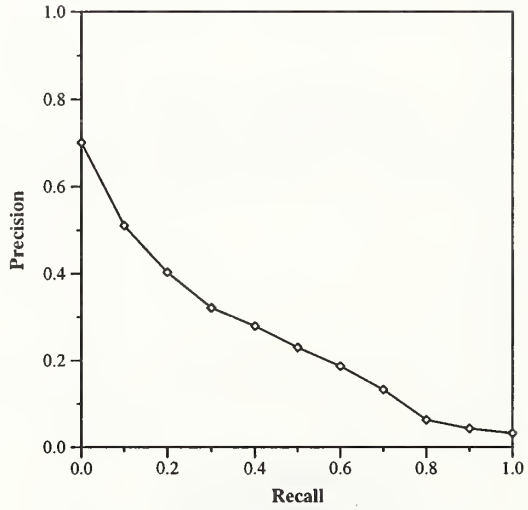
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2374

Recall Level Precision Averages	
Recall	Precision
0.00	0.7016
0.10	0.5107
0.20	0.4035
0.30	0.3216
0.40	0.2797
0.50	0.2305
0.60	0.1871
0.70	0.1330
0.80	0.0632
0.90	0.0432
1.00	0.0322

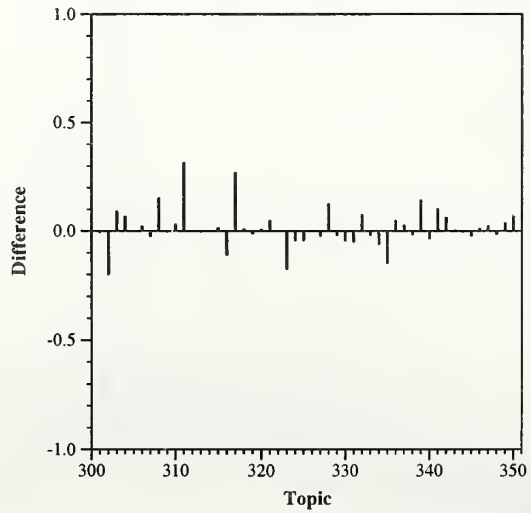
Average precision over all relevant docs	
non-interpolated	0.2389

Document Level Averages	
	Precision
At 5 docs	0.4640
At 10 docs	0.4220
At 15 docs	0.3653
At 20 docs	0.3370
At 30 docs	0.3033
At 100 docs	0.1928
At 200 docs	0.1282
At 500 docs	0.0790
At 1000 docs	0.0475

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2772



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	uwmt6a2
Run Description	Category A, Automatic, short
Number of Topics	50

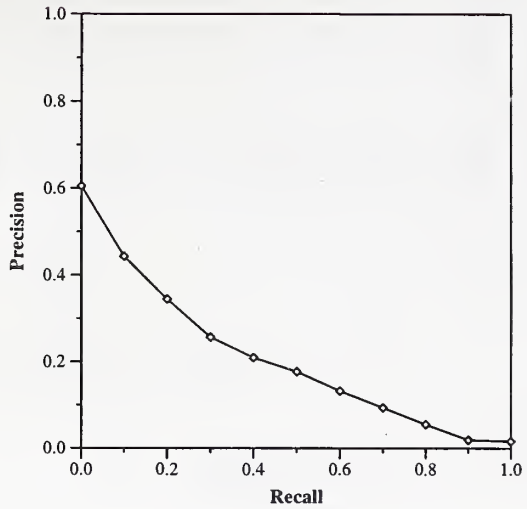
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1912

Recall Level Precision Averages	
Recall	Precision
0.00	0.6051
0.10	0.4433
0.20	0.3447
0.30	0.2567
0.40	0.2093
0.50	0.1769
0.60	0.1324
0.70	0.0940
0.80	0.0553
0.90	0.0194
1.00	0.0163

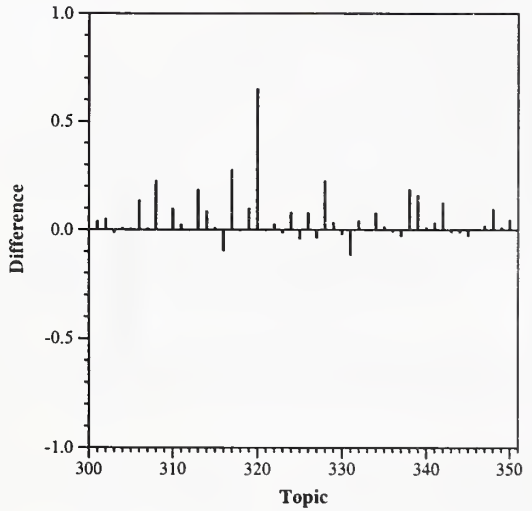
Average precision over all relevant docs	
non-interpolated	0.1912

Document Level Averages	
	Precision
At 5 docs	0.3960
At 10 docs	0.3540
At 15 docs	0.3173
At 20 docs	0.2940
At 30 docs	0.2567
At 100 docs	0.1614
At 200 docs	0.1098
At 500 docs	0.0626
At 1000 docs	0.0382

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2400



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	VrtyAH6a
Run Description	Category A, Automatic, short
Number of Topics	50

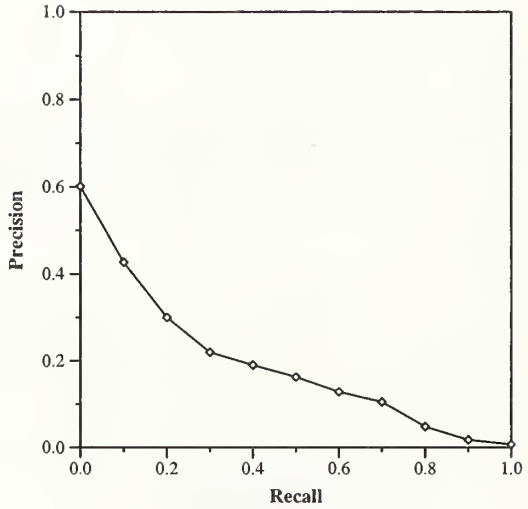
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1929

Recall Level Precision Averages	
Recall	Precision
0.00	0.6012
0.10	0.4270
0.20	0.3004
0.30	0.2202
0.40	0.1906
0.50	0.1630
0.60	0.1287
0.70	0.1053
0.80	0.0484
0.90	0.0179
1.00	0.0073

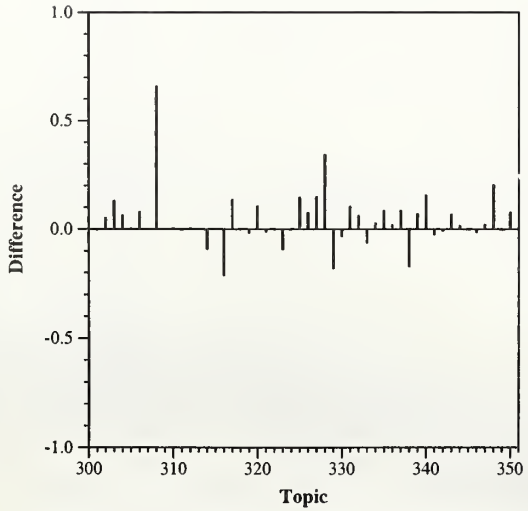
Average precision over all relevant docs	
non-interpolated	0.1777

Document Level Averages	
	Precision
At 5 docs	0.3600
At 10 docs	0.3060
At 15 docs	0.2720
At 20 docs	0.2640
At 30 docs	0.2393
At 100 docs	0.1602
At 200 docs	0.1093
At 500 docs	0.0609
At 1000 docs	0.0386

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2164



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	VrtyAH6b
Run Description	Category A, Automatic, long
Number of Topics	50

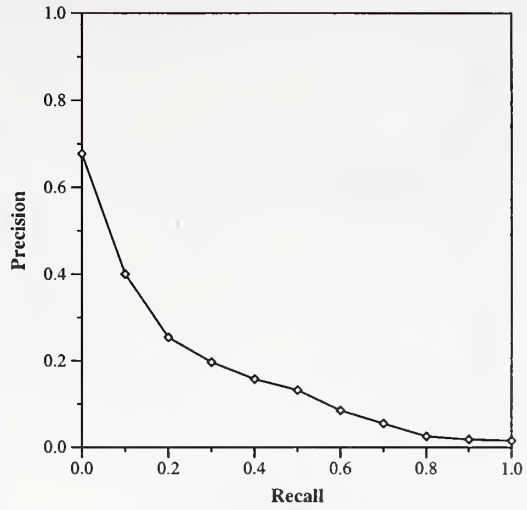
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2072

Recall Level Precision Averages	
Recall	Precision
0.00	0.6775
0.10	0.4008
0.20	0.2545
0.30	0.1972
0.40	0.1580
0.50	0.1326
0.60	0.0855
0.70	0.0553
0.80	0.0257
0.90	0.0187
1.00	0.0158

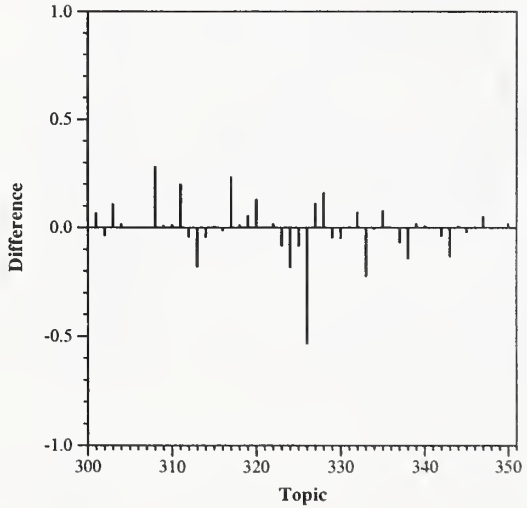
Average precision over all relevant docs	
non-interpolated	0.1542

Document Level Averages	
	Precision
At 5 docs	0.4160
At 10 docs	0.3420
At 15 docs	0.3107
At 20 docs	0.2820
At 30 docs	0.2493
At 100 docs	0.1498
At 200 docs	0.1060
At 500 docs	0.0640
At 1000 docs	0.0414

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2013



Recall-Precision Curve

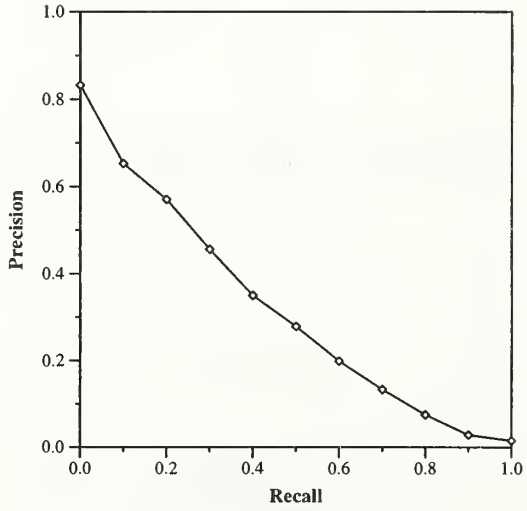


Difference from Median in Average Precision per Topic

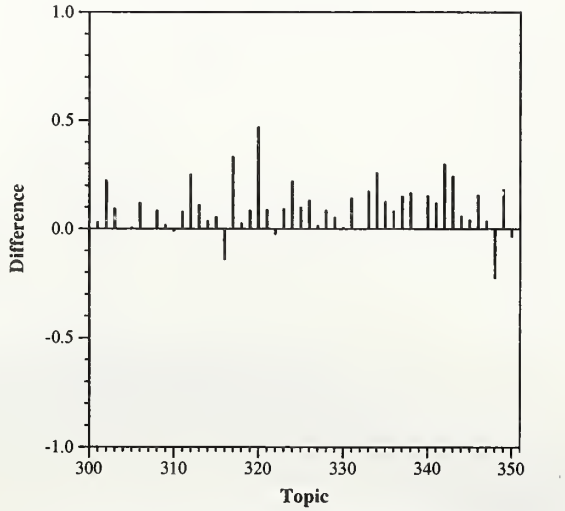
Summary Statistics	
Run Number	anu6min1
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	3042

Recall Level Precision Averages	
Recall	Precision
0.00	0.8325
0.10	0.6531
0.20	0.5710
0.30	0.4565
0.40	0.3505
0.50	0.2789
0.60	0.1991
0.70	0.1335
0.80	0.0751
0.90	0.0286
1.00	0.0154
Average precision over all relevant docs	
non-interpolated	0.3044

Document Level Averages	
	Precision
At 5 docs	0.6400
At 10 docs	0.5600
At 15 docs	0.5120
At 20 docs	0.4670
At 30 docs	0.4020
At 100 docs	0.2442
At 200 docs	0.1732
At 500 docs	0.0981
At 1000 docs	0.0608
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3427



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	CLAUG
Run Description	Category A, Manual
Number of Topics	50

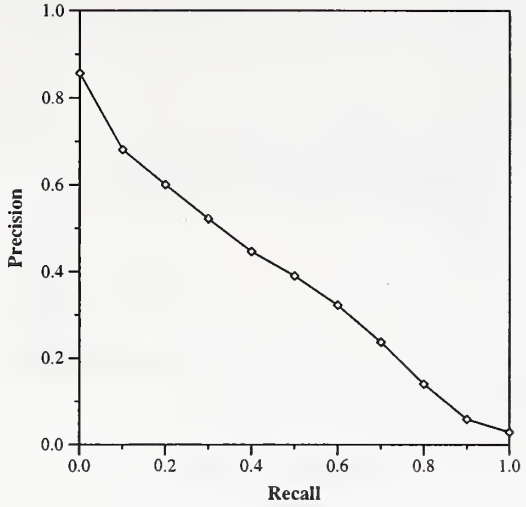
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	3095

Recall Level Precision Averages	
Recall	Precision
0.00	0.8567
0.10	0.6812
0.20	0.6007
0.30	0.5218
0.40	0.4464
0.50	0.3902
0.60	0.3226
0.70	0.2373
0.80	0.1407
0.90	0.0589
1.00	0.0296

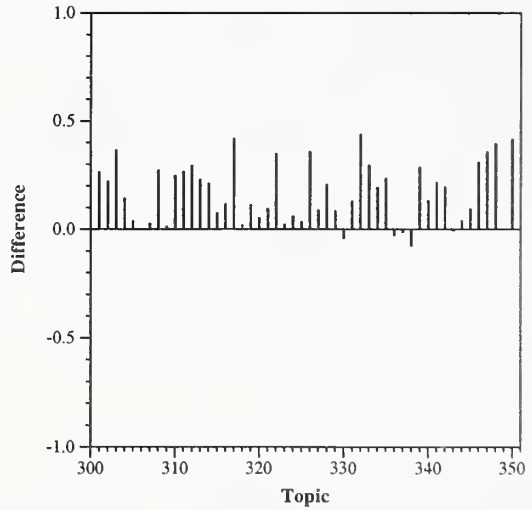
Average precision over all relevant docs	
non-interpolated	0.3742

Document Level Averages	
	Precision
At 5 docs	0.7120
At 10 docs	0.6120
At 15 docs	0.5493
At 20 docs	0.5080
At 30 docs	0.4620
At 100 docs	0.2822
At 200 docs	0.1898
At 500 docs	0.1037
At 1000 docs	0.0619

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3914



Recall-Precision Curve

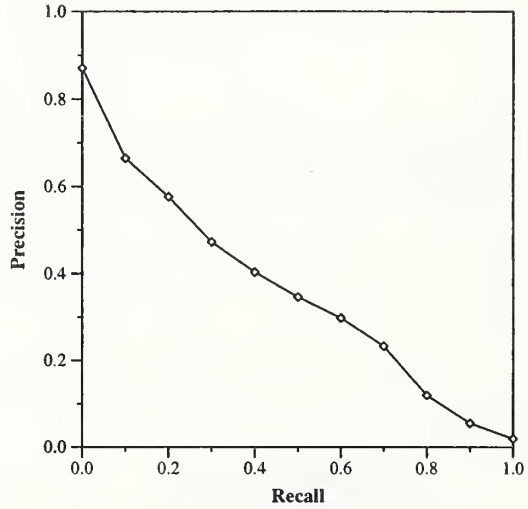


Difference from Median in Average Precision per Topic

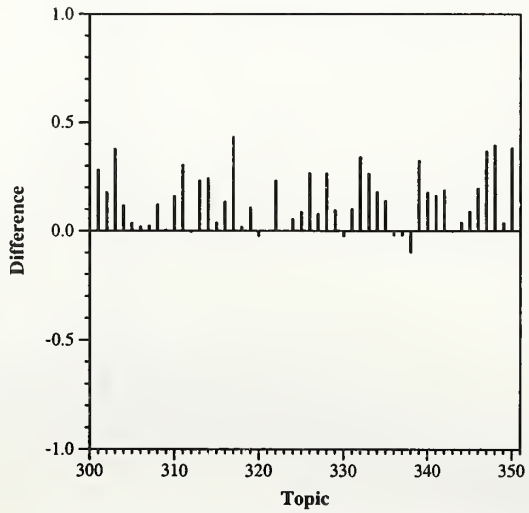
Summary Statistics	
Run Number	CLREL
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2998

Recall Level Precision Averages	
Recall	Precision
0.00	0.8711
0.10	0.6652
0.20	0.5765
0.30	0.4728
0.40	0.4036
0.50	0.3463
0.60	0.2979
0.70	0.2332
0.80	0.1200
0.90	0.0555
1.00	0.0195
Average precision over all relevant docs	
non-interpolated	0.3514

Document Level Averages	
	Precision
At 5 docs	0.7000
At 10 docs	0.5960
At 15 docs	0.5280
At 20 docs	0.4910
At 30 docs	0.4340
At 100 docs	0.2712
At 200 docs	0.1832
At 500 docs	0.1006
At 1000 docs	0.0600
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3639



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	fsclt6
Run Description	Category A, Manual
Number of Topics	50

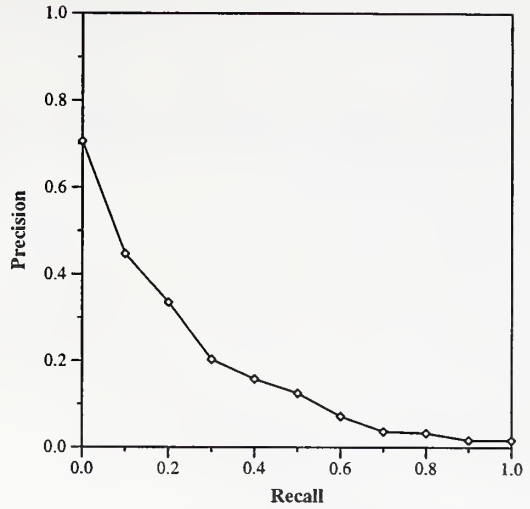
Total number of documents over all topics	
Retrieved:	12119
Relevant:	4611
Rel-ret:	1300

Recall Level Precision Averages	
Recall	Precision
0.00	0.7064
0.10	0.4472
0.20	0.3356
0.30	0.2034
0.40	0.1581
0.50	0.1255
0.60	0.0719
0.70	0.0370
0.80	0.0333
0.90	0.0167
1.00	0.0167

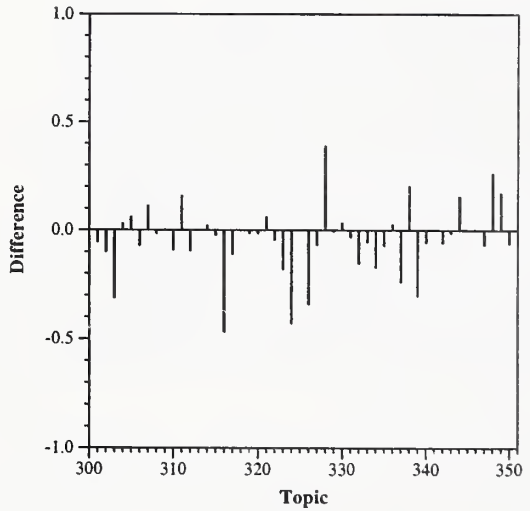
Average precision over all relevant docs	
non-interpolated	0.1691

Document Level Averages	
	Precision
At 5 docs	0.4520
At 10 docs	0.3900
At 15 docs	0.3467
At 20 docs	0.3120
At 30 docs	0.2680
At 100 docs	0.1642
At 200 docs	0.1037
At 500 docs	0.0500
At 1000 docs	0.0260

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2173



Recall-Precision Curve

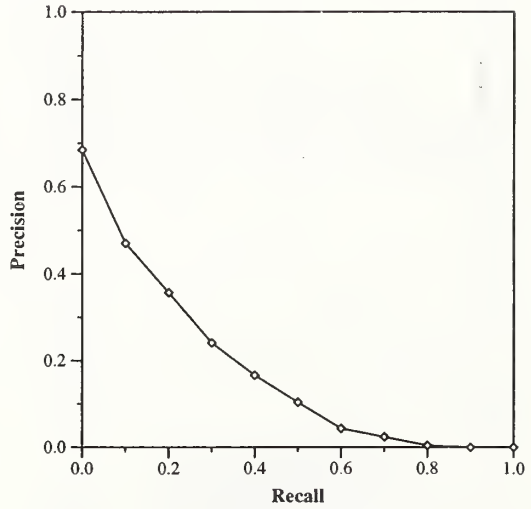


Difference from Median in Average Precision per Topic

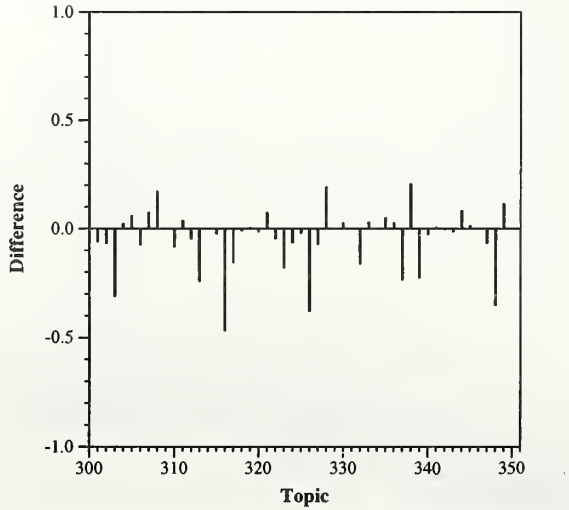
Summary Statistics	
Run Number	fsclt6r
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	49001
Relevant:	4611
Rel-ret:	1659

Recall Level Precision Averages	
Recall	Precision
0.00	0.6847
0.10	0.4703
0.20	0.3571
0.30	0.2414
0.40	0.1665
0.50	0.1040
0.60	0.0436
0.70	0.0239
0.80	0.0040
0.90	0.0000
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.1660

Document Level Averages	
	Precision
At 5 docs	0.4280
At 10 docs	0.3800
At 15 docs	0.3400
At 20 docs	0.3150
At 30 docs	0.2800
At 100 docs	0.1682
At 200 docs	0.1121
At 500 docs	0.0570
At 1000 docs	0.0332
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2142



Recall-Precision Curve

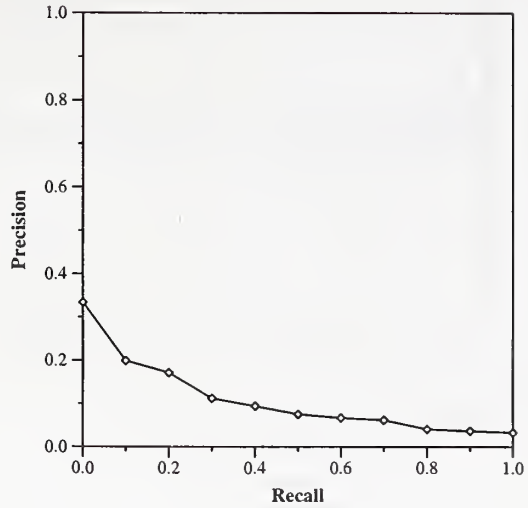


Difference from Median in Average Precision per Topic

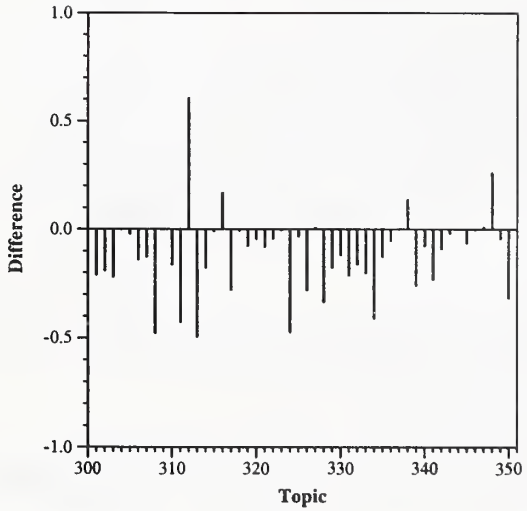
Summary Statistics	
Run Number	fsclt6t
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	43156
Relevant:	4611
Rel-ret:	1172

Recall Level Precision Averages	
Recall	Precision
0.00	0.3340
0.10	0.1992
0.20	0.1718
0.30	0.1125
0.40	0.0943
0.50	0.0756
0.60	0.0670
0.70	0.0620
0.80	0.0410
0.90	0.0367
1.00	0.0334
Average precision over all relevant docs	
non-interpolated	0.0958

Document Level Averages	
	Precision
At 5 docs	0.1720
At 10 docs	0.1680
At 15 docs	0.1533
At 20 docs	0.1430
At 30 docs	0.1233
At 100 docs	0.0752
At 200 docs	0.0531
At 500 docs	0.0336
At 1000 docs	0.0234
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1279



Recall-Precision Curve

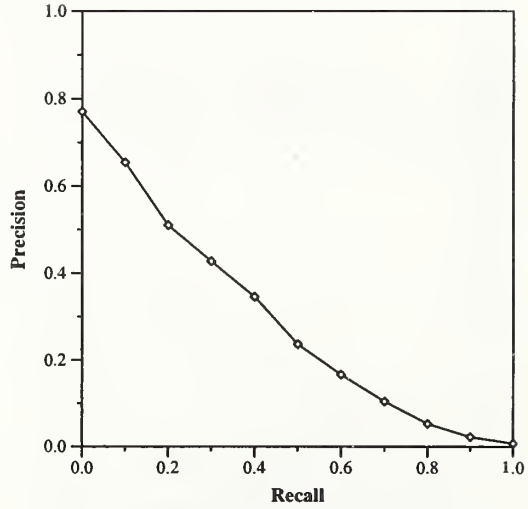


Difference from Median in Average Precision per Topic

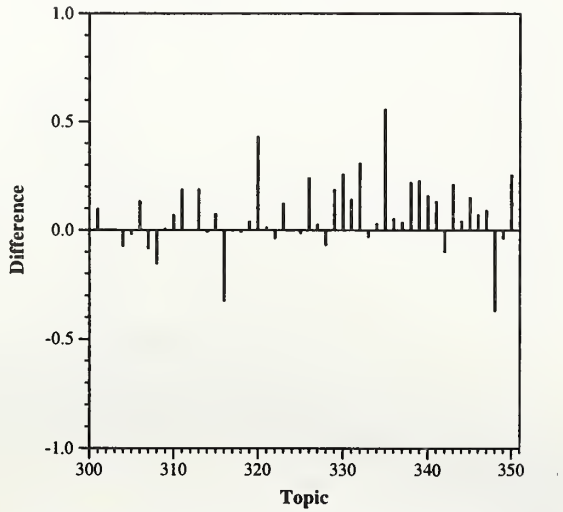
Summary Statistics	
Run Number	gerua1
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2669

Recall Level Precision Averages	
Recall	Precision
0.00	0.7707
0.10	0.6545
0.20	0.5105
0.30	0.4275
0.40	0.3460
0.50	0.2366
0.60	0.1666
0.70	0.1046
0.80	0.0529
0.90	0.0225
1.00	0.0076
Average precision over all relevant docs	
non-interpolated	0.2783

Document Level Averages	
	Precision
At 5 docs	0.5760
At 10 docs	0.5200
At 15 docs	0.4680
At 20 docs	0.4400
At 30 docs	0.3933
At 100 docs	0.2598
At 200 docs	0.1704
At 500 docs	0.0901
At 1000 docs	0.0534
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3241



Recall-Precision Curve

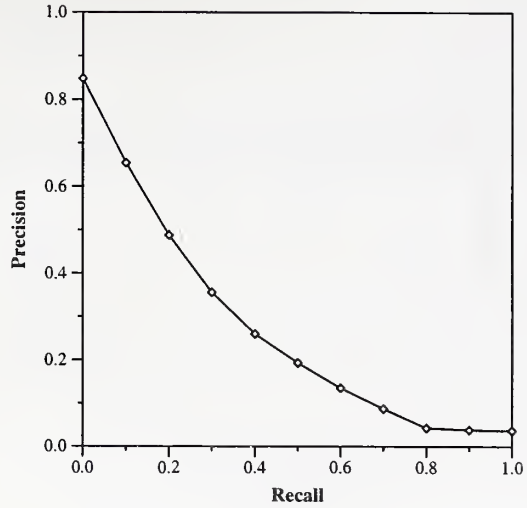


Difference from Median in Average Precision per Topic

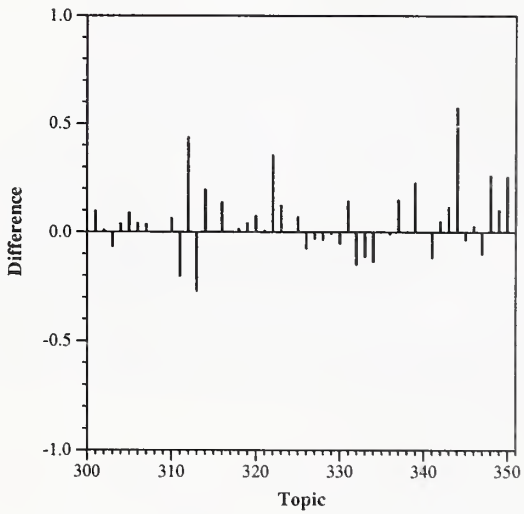
Summary Statistics	
Run Number	gerua2
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	32086
Relevant:	4611
Rel-ret:	2161

Recall Level Precision Averages	
Recall	Precision
0.00	0.8484
0.10	0.6547
0.20	0.4882
0.30	0.3560
0.40	0.2602
0.50	0.1936
0.60	0.1353
0.70	0.0875
0.80	0.0431
0.90	0.0387
1.00	0.0369
Average precision over all relevant docs	
non-interpolated	0.2559

Document Level Averages	
	Precision
At 5 docs	0.6160
At 10 docs	0.5480
At 15 docs	0.4867
At 20 docs	0.4290
At 30 docs	0.3593
At 100 docs	0.1968
At 200 docs	0.1311
At 500 docs	0.0723
At 1000 docs	0.0432
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3022



Recall-Precision Curve

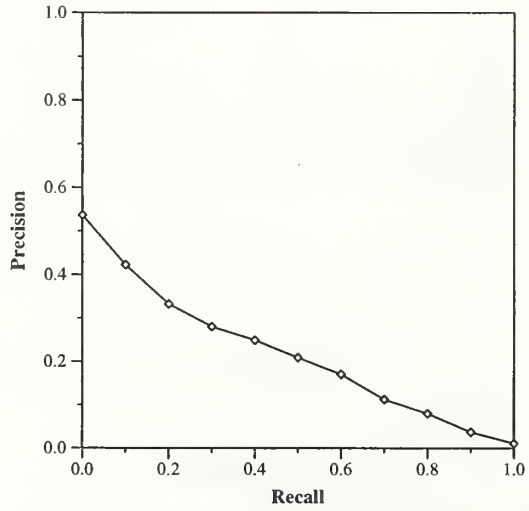


Difference from Median in Average Precision per Topic

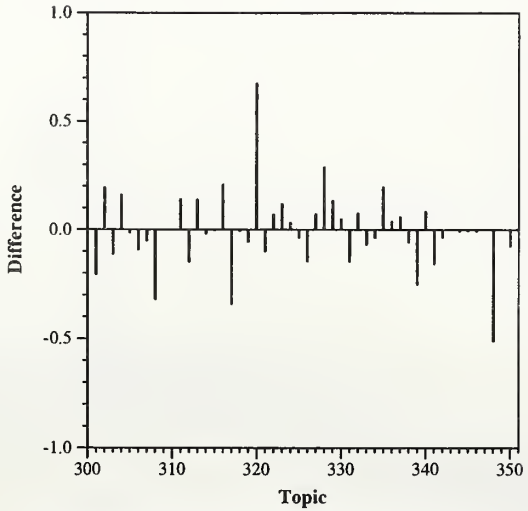
Summary Statistics	
Run Number	gmu97ma1
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	27386
Relevant:	4611
Rel-ret:	2228

Recall Level Precision Averages	
Recall	Precision
0.00	0.5367
0.10	0.4225
0.20	0.3327
0.30	0.2806
0.40	0.2496
0.50	0.2092
0.60	0.1706
0.70	0.1128
0.80	0.0802
0.90	0.0373
1.00	0.0112
Average precision over all relevant docs	
non-interpolated	0.2041

Document Level Averages	
	Precision
At 5 docs	0.3280
At 10 docs	0.3000
At 15 docs	0.3080
At 20 docs	0.2980
At 30 docs	0.2720
At 100 docs	0.1938
At 200 docs	0.1347
At 500 docs	0.0748
At 1000 docs	0.0446
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2417



Recall-Precision Curve

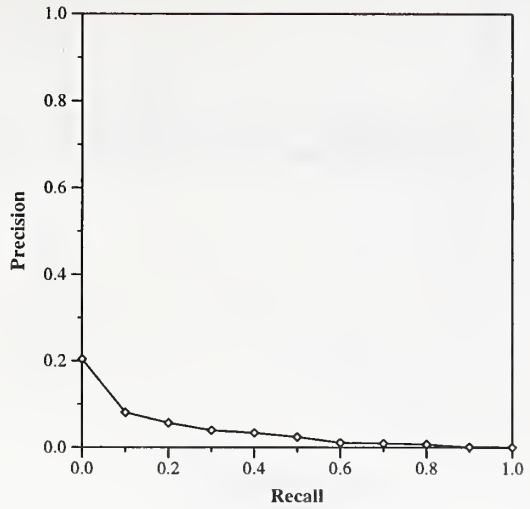


Difference from Median in Average Precision per Topic

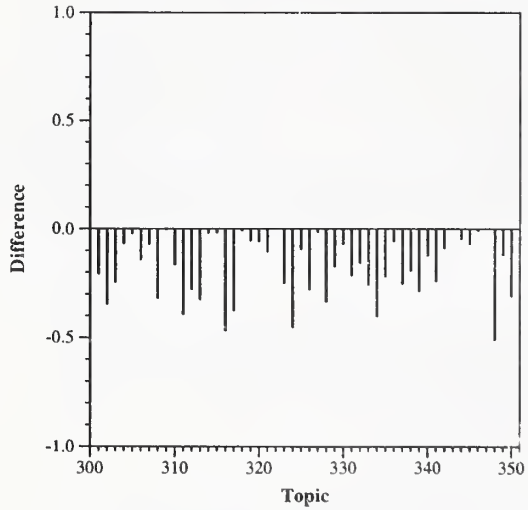
Summary Statistics	
Run Number	gmu97ma2
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	46473
Relevant:	4611
Rel-ret:	1156

Recall Level Precision Averages	
Recall	Precision
0.00	0.2043
0.10	0.0810
0.20	0.0569
0.30	0.0396
0.40	0.0338
0.50	0.0244
0.60	0.0111
0.70	0.0093
0.80	0.0070
0.90	0.0012
1.00	0.0007
Average precision over all relevant docs	
non-interpolated	0.0320

Document Level Averages	
	Precision
At 5 docs	0.0680
At 10 docs	0.0580
At 15 docs	0.0680
At 20 docs	0.0710
At 30 docs	0.0640
At 100 docs	0.0492
At 200 docs	0.0421
At 500 docs	0.0321
At 1000 docs	0.0231
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0533



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	harris1
Run Description	Category A, Manual
Number of Topics	50

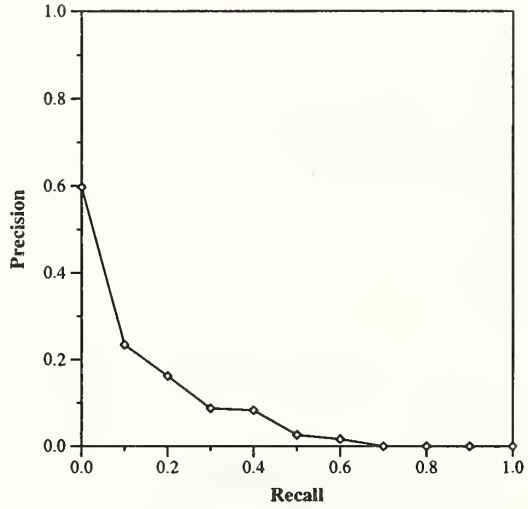
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	725

Recall Level Precision Averages	
Recall	Precision
0.00	0.5972
0.10	0.2345
0.20	0.1624
0.30	0.0877
0.40	0.0833
0.50	0.0262
0.60	0.0164
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000

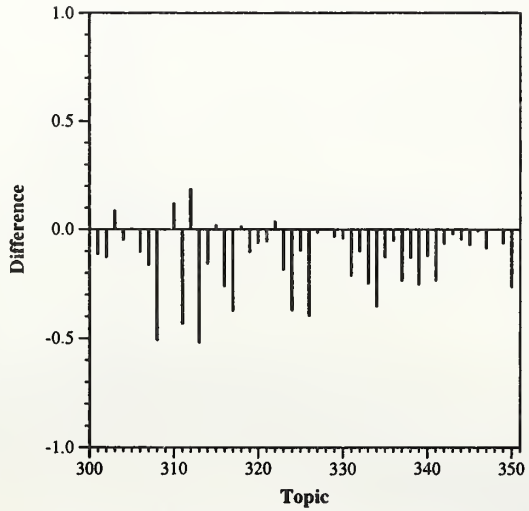
Average precision over all relevant docs	
non-interpolated	0.0821

Document Level Averages	
	Precision
At 5 docs	0.4040
At 10 docs	0.2880
At 15 docs	0.2360
At 20 docs	0.2000
At 30 docs	0.1500
At 100 docs	0.0698
At 200 docs	0.0437
At 500 docs	0.0238
At 1000 docs	0.0145

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1085



Recall-Precision Curve

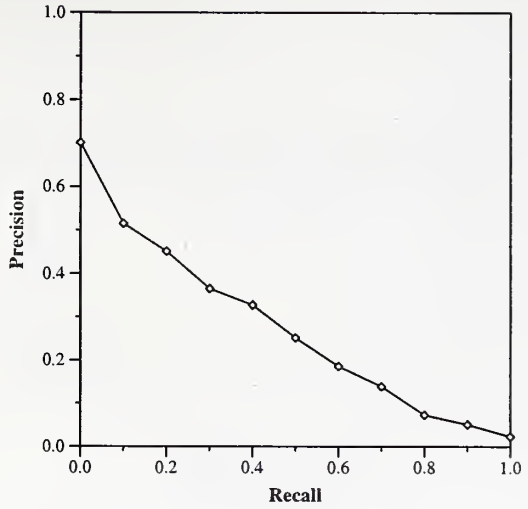


Difference from Median in Average Precision per Topic

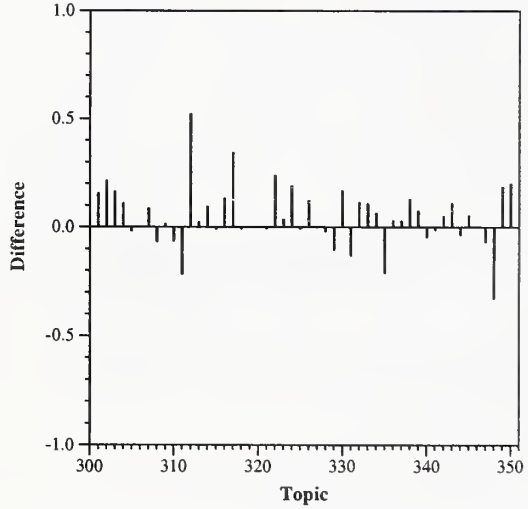
Summary Statistics	
Run Number	iss97man
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	46964
Relevant:	4611
Rel-ret:	2792

Recall Level Precision Averages	
Recall	Precision
0.00	0.7014
0.10	0.5155
0.20	0.4512
0.30	0.3653
0.40	0.3275
0.50	0.2517
0.60	0.1859
0.70	0.1393
0.80	0.0734
0.90	0.0513
1.00	0.0235
Average precision over all relevant docs	
non-interpolated	0.2576

Document Level Averages	
	Precision
At 5 docs	0.4400
At 10 docs	0.4120
At 15 docs	0.3587
At 20 docs	0.3440
At 30 docs	0.3147
At 100 docs	0.2198
At 200 docs	0.1554
At 500 docs	0.0921
At 1000 docs	0.0558
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2975



Recall-Precision Curve

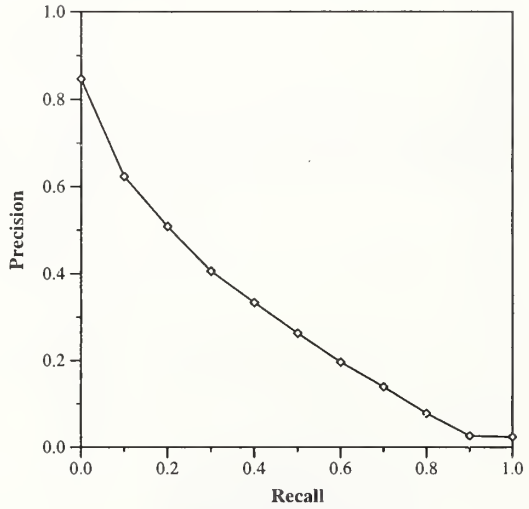


Difference from Median in Average Precision per Topic

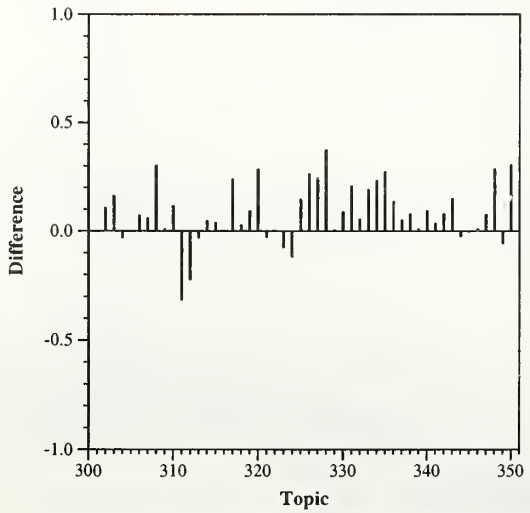
Summary Statistics	
Run Number	LNmShort
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2849

Recall Level Precision Averages	
Recall	Precision
0.00	0.8471
0.10	0.6236
0.20	0.5085
0.30	0.4061
0.40	0.3341
0.50	0.2632
0.60	0.1966
0.70	0.1397
0.80	0.0784
0.90	0.0264
1.00	0.0240
Average precision over all relevant docs	
non-interpolated	0.2902

Document Level Averages	
	Precision
At 5 docs	0.6120
At 10 docs	0.5000
At 15 docs	0.4547
At 20 docs	0.4160
At 30 docs	0.3547
At 100 docs	0.2196
At 200 docs	0.1494
At 500 docs	0.0891
At 1000 docs	0.0570
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3188



Recall-Precision Curve

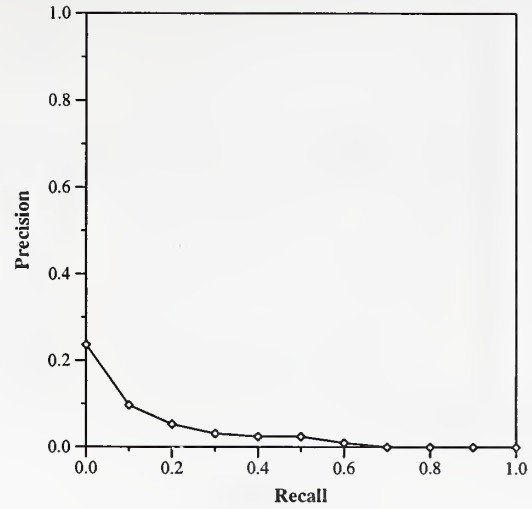


Difference from Median in Average Precision per Topic

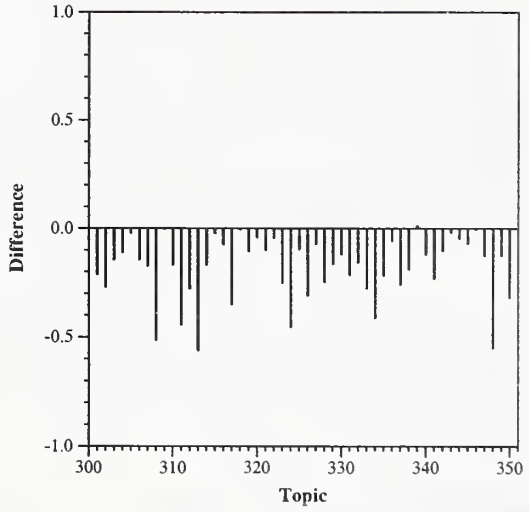
Summary Statistics	
Run Number	nmsu2
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	11269
Relevant:	4611
Rel-ret:	203

Recall Level Precision Averages	
Recall	Precision
0.00	0.2364
0.10	0.0968
0.20	0.0528
0.30	0.0312
0.40	0.0244
0.50	0.0244
0.60	0.0095
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.0271

Document Level Averages	
	Precision
At 5 docs	0.1080
At 10 docs	0.0860
At 15 docs	0.0680
At 20 docs	0.0590
At 30 docs	0.0507
At 100 docs	0.0248
At 200 docs	0.0131
At 500 docs	0.0070
At 1000 docs	0.0041
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0431



Recall-Precision Curve

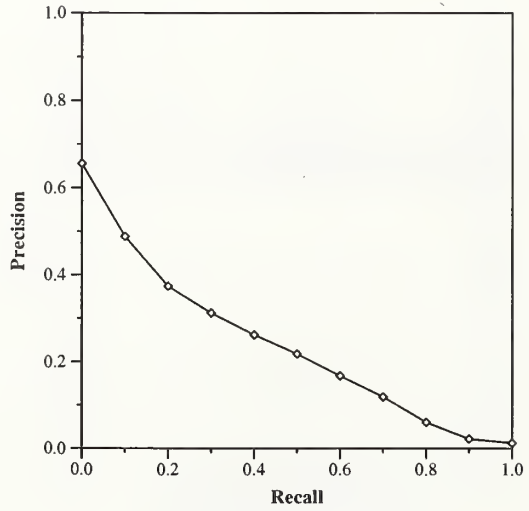


Difference from Median in Average Precision per Topic

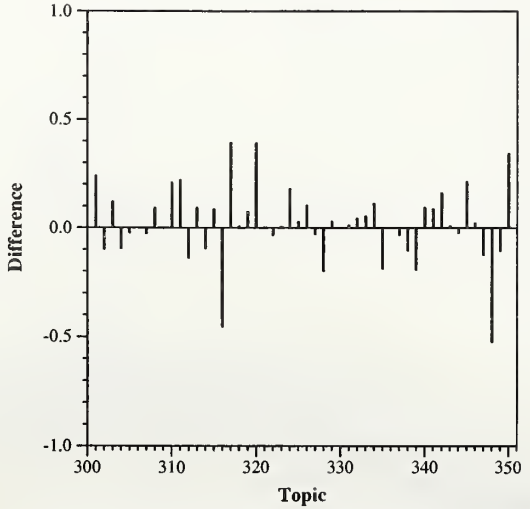
Summary Statistics	
Run Number	Brkly23
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	2583

Recall Level Precision Averages	
Recall	Precision
0.00	0.6558
0.10	0.4885
0.20	0.3745
0.30	0.3128
0.40	0.2623
0.50	0.2182
0.60	0.1677
0.70	0.1193
0.80	0.0604
0.90	0.0221
1.00	0.0126
Average precision over all relevant docs	
non-interpolated	0.2282

Document Level Averages	
	Precision
At 5 docs	0.4880
At 10 docs	0.4320
At 15 docs	0.3813
At 20 docs	0.3510
At 30 docs	0.3140
At 100 docs	0.2112
At 200 docs	0.1433
At 500 docs	0.0816
At 1000 docs	0.0517
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2612



Recall-Precision Curve

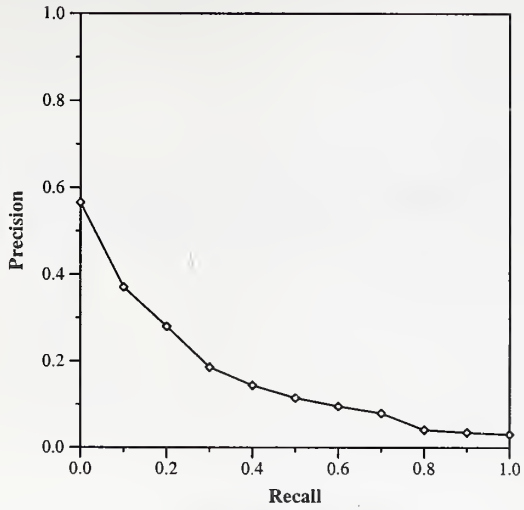


Difference from Median in Average Precision per Topic

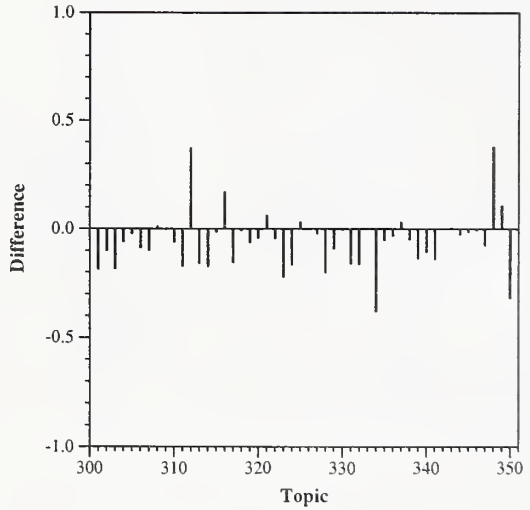
Summary Statistics	
Run Number	glair62
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	1831

Recall Level Precision Averages	
Recall	Precision
0.00	0.5656
0.10	0.3708
0.20	0.2800
0.30	0.1860
0.40	0.1440
0.50	0.1148
0.60	0.0955
0.70	0.0795
0.80	0.0410
0.90	0.0351
1.00	0.0306
Average precision over all relevant docs	
non-interpolated	0.1536

Document Level Averages	
	Precision
At 5 docs	0.3320
At 10 docs	0.2900
At 15 docs	0.2733
At 20 docs	0.2550
At 30 docs	0.2287
At 100 docs	0.1266
At 200 docs	0.0888
At 500 docs	0.0576
At 1000 docs	0.0366
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1890



Recall-Precision Curve

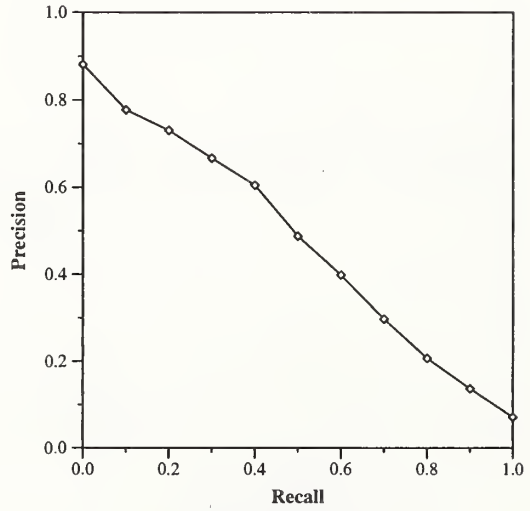


Difference from Median in Average Precision per Topic

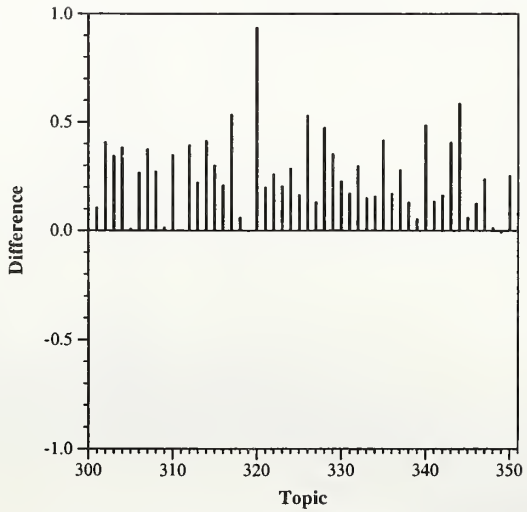
Summary Statistics	
Run Number	uwmt6a0
Run Description	Category A, Manual
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4611
Rel-ret:	3058

Recall Level Precision Averages	
Recall	Precision
0.00	0.8812
0.10	0.7778
0.20	0.7312
0.30	0.6671
0.40	0.6052
0.50	0.4880
0.60	0.3989
0.70	0.2977
0.80	0.2070
0.90	0.1371
1.00	0.0715
Average precision over all relevant docs	
non-interpolated	0.4630

Document Level Averages	
	Precision
At 5 docs	0.7080
At 10 docs	0.6820
At 15 docs	0.6387
At 20 docs	0.6040
At 30 docs	0.5460
At 100 docs	0.3578
At 200 docs	0.2217
At 500 docs	0.1096
At 1000 docs	0.0612
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4896



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	jhuapln
Run Description	Category B, Automatic, short
Number of Topics	47

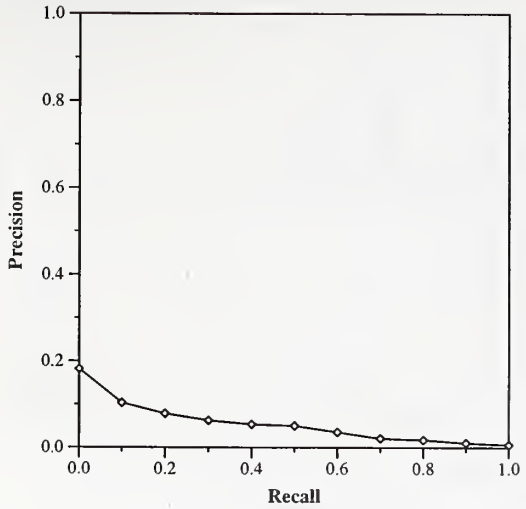
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	511

Recall Level Precision Averages	
Recall	Precision
0.00	0.1817
0.10	0.1035
0.20	0.0788
0.30	0.0630
0.40	0.0537
0.50	0.0505
0.60	0.0359
0.70	0.0212
0.80	0.0178
0.90	0.0108
1.00	0.0068

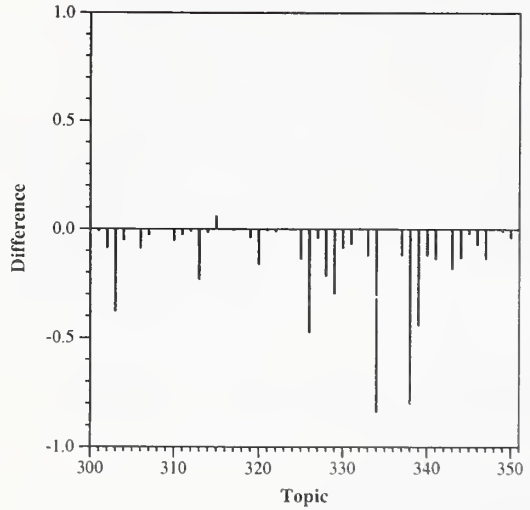
Average precision over all relevant docs	
non-interpolated	0.0477

Document Level Averages	
	Precision
At 5 docs	0.0681
At 10 docs	0.0766
At 15 docs	0.0780
At 20 docs	0.0734
At 30 docs	0.0624
At 100 docs	0.0385
At 200 docs	0.0271
At 500 docs	0.0169
At 1000 docs	0.0109

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0508



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	jhuapls
Run Description	Category B, Automatic, short
Number of Topics	47

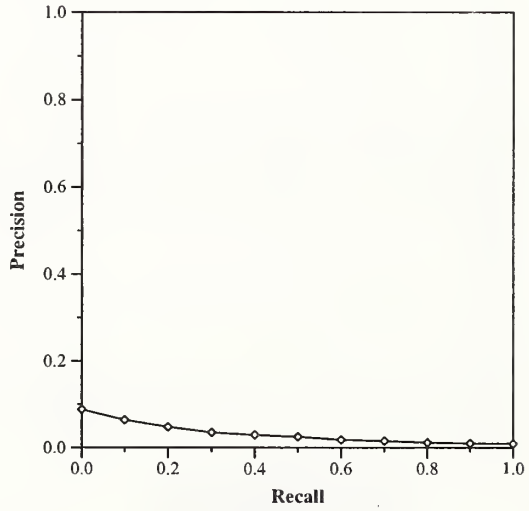
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	484

Recall Level Precision Averages	
Recall	Precision
0.00	0.0884
0.10	0.0642
0.20	0.0478
0.30	0.0350
0.40	0.0293
0.50	0.0253
0.60	0.0185
0.70	0.0157
0.80	0.0120
0.90	0.0100
1.00	0.0096

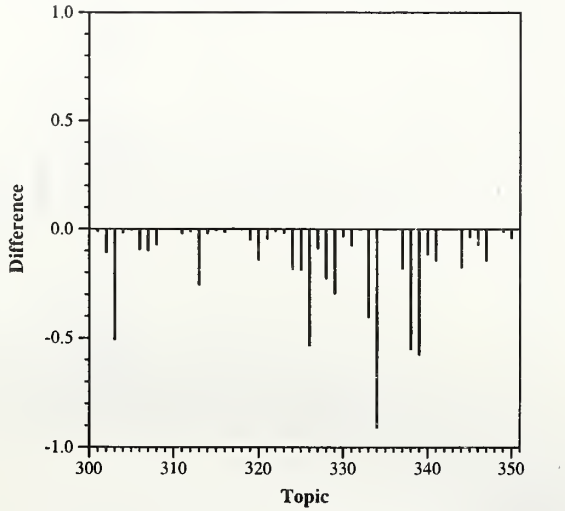
Average precision over all relevant docs	
non-interpolated	0.0288

Document Level Averages	
	Precision
At 5 docs	0.0426
At 10 docs	0.0404
At 15 docs	0.0312
At 20 docs	0.0330
At 30 docs	0.0277
At 100 docs	0.0213
At 200 docs	0.0177
At 500 docs	0.0130
At 1000 docs	0.0103

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0399



Recall-Precision Curve

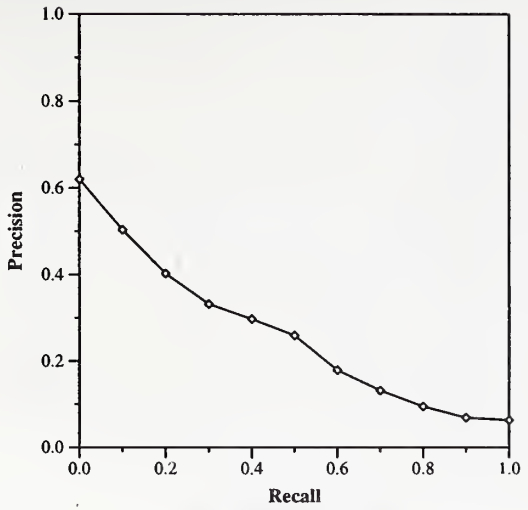


Difference from Median in Average Precision per Topic

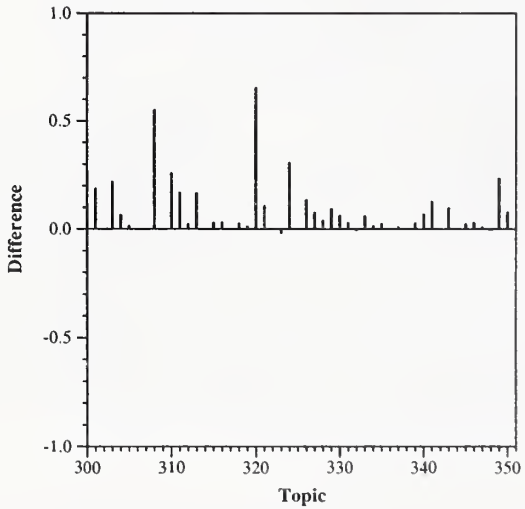
Summary Statistics	
Run Number	unc6aal
Run Description	Category B, Automatic, long
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	1024

Recall Level Precision Averages	
Recall	Precision
0.00	0.6200
0.10	0.5036
0.20	0.4028
0.30	0.3325
0.40	0.2975
0.50	0.2596
0.60	0.1794
0.70	0.1326
0.80	0.0953
0.90	0.0693
1.00	0.0638
Average precision over all relevant docs	
non-interpolated	0.2518

Document Level Averages	
	Precision
At 5 docs	0.3702
At 10 docs	0.3064
At 15 docs	0.2709
At 20 docs	0.2340
At 30 docs	0.1972
At 100 docs	0.1045
At 200 docs	0.0693
At 500 docs	0.0370
At 1000 docs	0.0218
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2684



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	unc6aas
Run Description	Category B, Automatic, short
Number of Topics	47

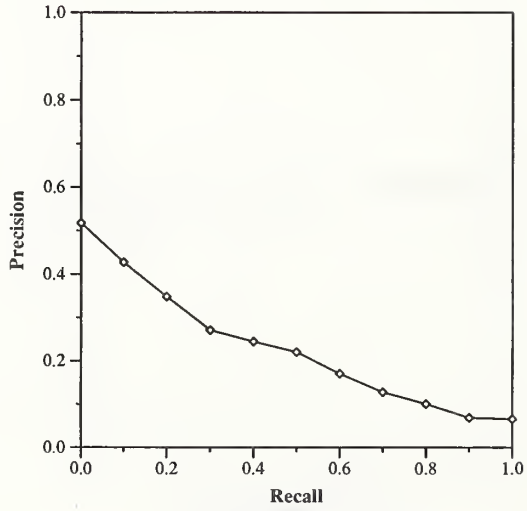
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	905

Recall Level Precision Averages	
Recall	Precision
0.00	0.5176
0.10	0.4271
0.20	0.3484
0.30	0.2709
0.40	0.2446
0.50	0.2207
0.60	0.1706
0.70	0.1283
0.80	0.1008
0.90	0.0690
1.00	0.0658

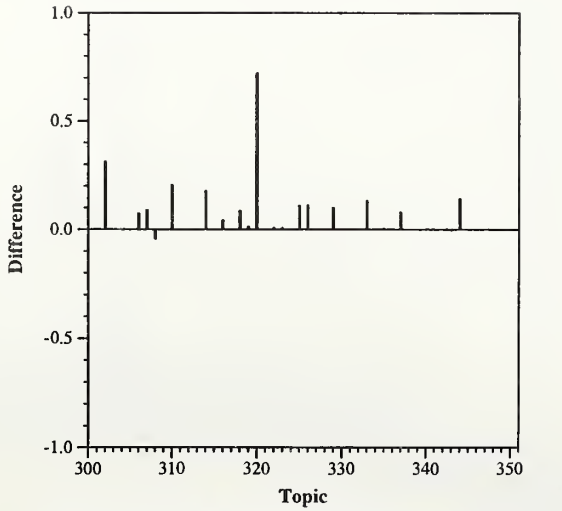
Average precision over all relevant docs	
non-interpolated	0.2167

Document Level Averages	
	Precision
At 5 docs	0.3234
At 10 docs	0.2766
At 15 docs	0.2496
At 20 docs	0.2138
At 30 docs	0.1738
At 100 docs	0.0913
At 200 docs	0.0585
At 500 docs	0.0327
At 1000 docs	0.0193

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2378



Recall-Precision Curve

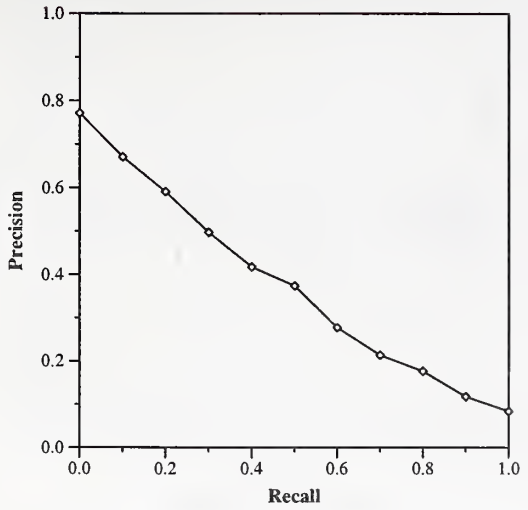


Difference from Median in Average Precision per Topic

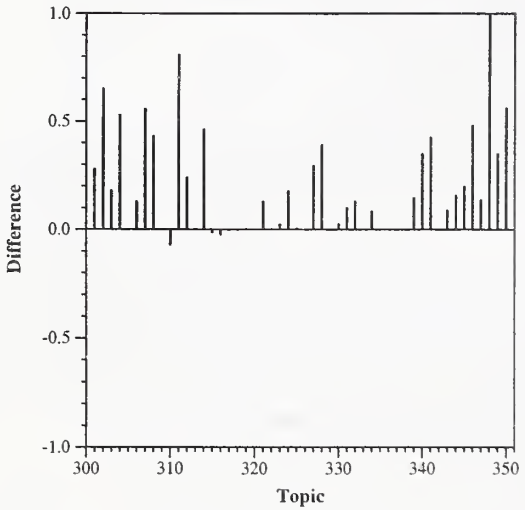
Summary Statistics	
Run Number	unc6ma
Run Description	Category B, Manual
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	1143

Recall Level Precision Averages	
Recall	Precision
0.00	0.7717
0.10	0.6708
0.20	0.5907
0.30	0.4973
0.40	0.4175
0.50	0.3739
0.60	0.2777
0.70	0.2138
0.80	0.1767
0.90	0.1180
1.00	0.0842
Average precision over all relevant docs	
non-interpolated	0.3663

Document Level Averages	
	Precision
At 5 docs	0.5404
At 10 docs	0.4277
At 15 docs	0.3617
At 20 docs	0.3309
At 30 docs	0.2794
At 100 docs	0.1502
At 200 docs	0.0921
At 500 docs	0.0437
At 1000 docs	0.0243
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3776



Recall-Precision Curve

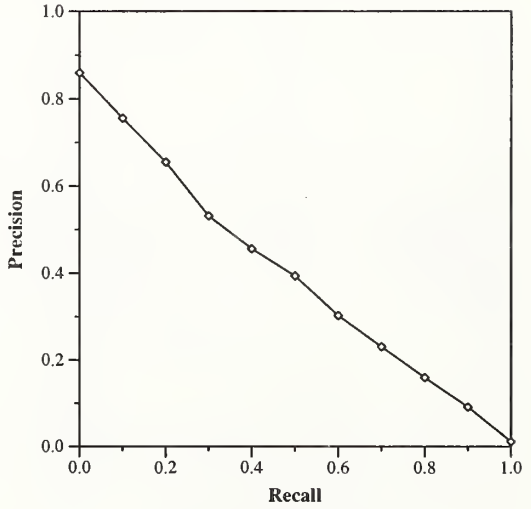


Difference from Median in Average Precision per Topic

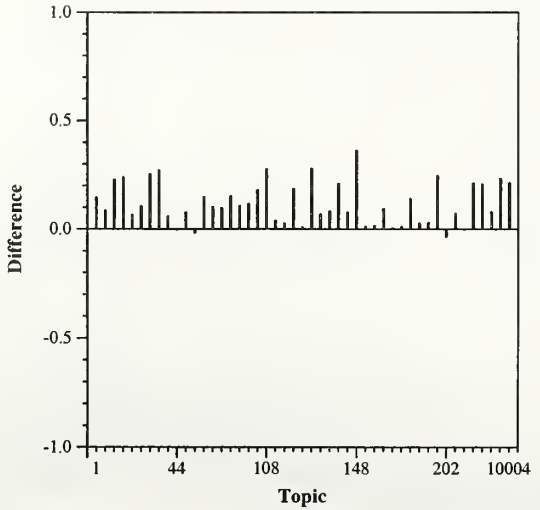
Summary Statistics	
Run Number	att97rc
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5339

Recall Level Precision Averages	
Recall	Precision
0.00	0.8600
0.10	0.7561
0.20	0.6556
0.30	0.5315
0.40	0.4562
0.50	0.3937
0.60	0.3027
0.70	0.2309
0.80	0.1596
0.90	0.0916
1.00	0.0119
Average precision over all relevant docs	
non-interpolated	0.3963

Document Level Averages	
	Precision
At 5 docs	0.7021
At 10 docs	0.6511
At 15 docs	0.6128
At 20 docs	0.5883
At 30 docs	0.5411
At 100 docs	0.4011
At 200 docs	0.3057
At 500 docs	0.1857
At 1000 docs	0.1136
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4060



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	att97re
Run Description	Category A
Number of Topics	47

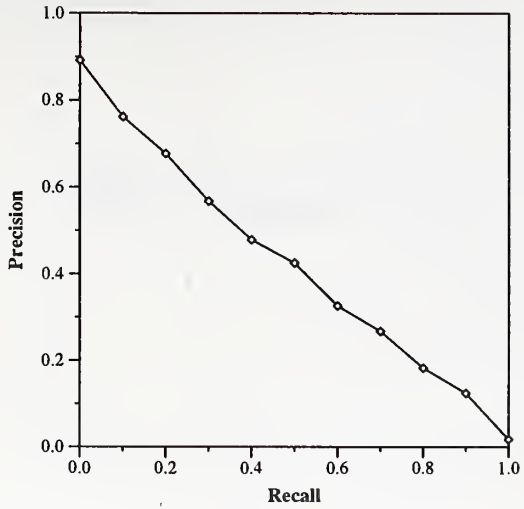
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5599

Recall Level Precision Averages	
Recall	Precision
0.00	0.8920
0.10	0.7623
0.20	0.6776
0.30	0.5678
0.40	0.4792
0.50	0.4254
0.60	0.3265
0.70	0.2676
0.80	0.1830
0.90	0.1246
1.00	0.0177

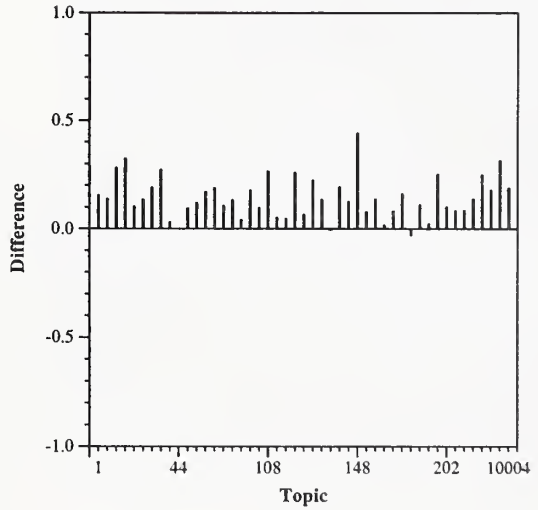
Average precision over all relevant docs	
non-interpolated	0.4207

Document Level Averages	
	Precision
At 5 docs	0.7149
At 10 docs	0.6723
At 15 docs	0.6355
At 20 docs	0.5968
At 30 docs	0.5504
At 100 docs	0.4183
At 200 docs	0.3185
At 500 docs	0.1946
At 1000 docs	0.1191

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4302



Recall-Precision Curve

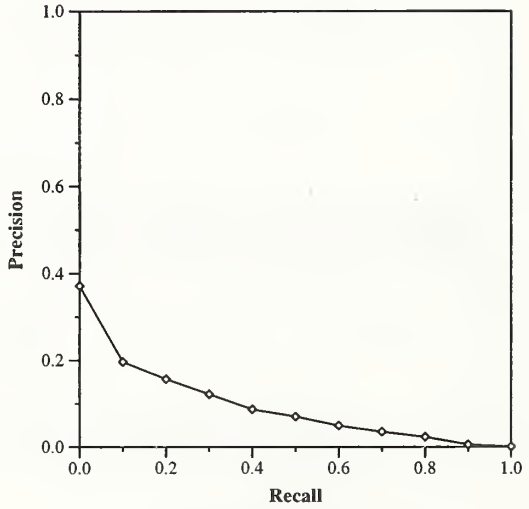


Difference from Median in Average Precision per Topic

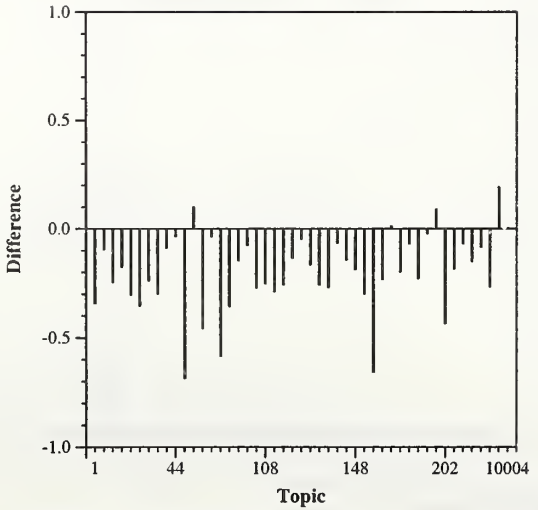
Summary Statistics	
Run Number	cir6rou1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	2533

Recall Level Precision Averages	
Recall	Precision
0.00	0.3714
0.10	0.1968
0.20	0.1571
0.30	0.1223
0.40	0.0870
0.50	0.0704
0.60	0.0493
0.70	0.0352
0.80	0.0232
0.90	0.0057
1.00	0.0009
Average precision over all relevant docs	
non-interpolated	0.0792

Document Level Averages	
	Precision
At 5 docs	0.1660
At 10 docs	0.1660
At 15 docs	0.1617
At 20 docs	0.1500
At 30 docs	0.1404
At 100 docs	0.1406
At 200 docs	0.1148
At 500 docs	0.0729
At 1000 docs	0.0539
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1317



Recall-Precision Curve

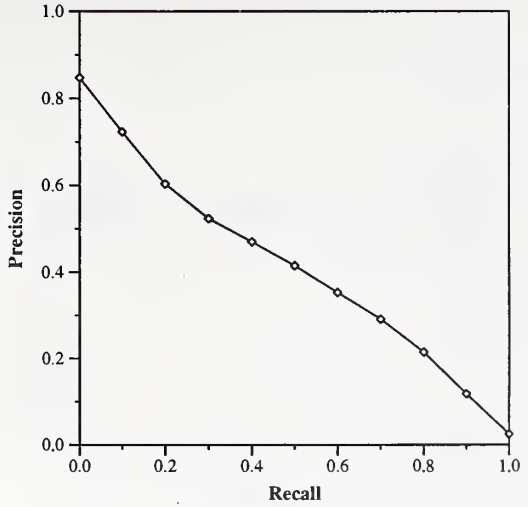


Difference from Median in Average Precision per Topic

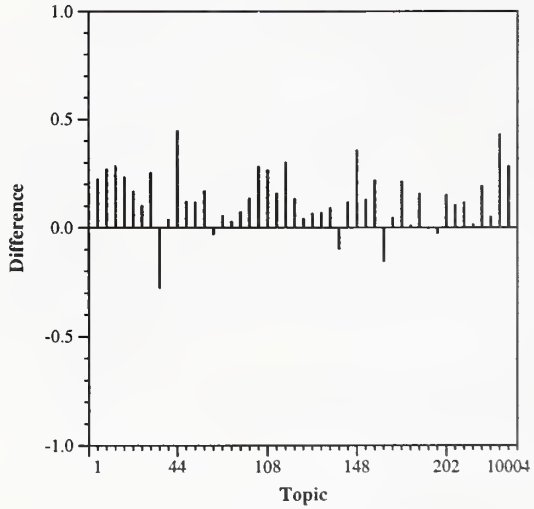
Summary Statistics	
Run Number	city6r1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5558

Recall Level Precision Averages	
Recall	Precision
0.00	0.8478
0.10	0.7235
0.20	0.6037
0.30	0.5233
0.40	0.4698
0.50	0.4149
0.60	0.3529
0.70	0.2911
0.80	0.2149
0.90	0.1182
1.00	0.0253
Average precision over all relevant docs	
non-interpolated	0.4076

Document Level Averages	
	Precision
At 5 docs	0.6979
At 10 docs	0.6532
At 15 docs	0.6057
At 20 docs	0.5777
At 30 docs	0.5475
At 100 docs	0.4104
At 200 docs	0.3204
At 500 docs	0.1940
At 1000 docs	0.1183
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4115



Recall-Precision Curve

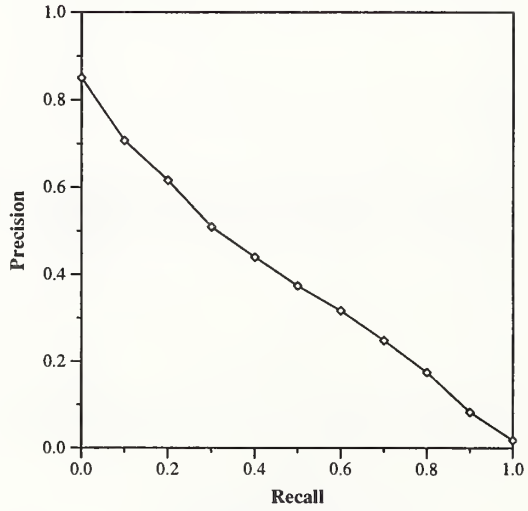


Difference from Median in Average Precision per Topic

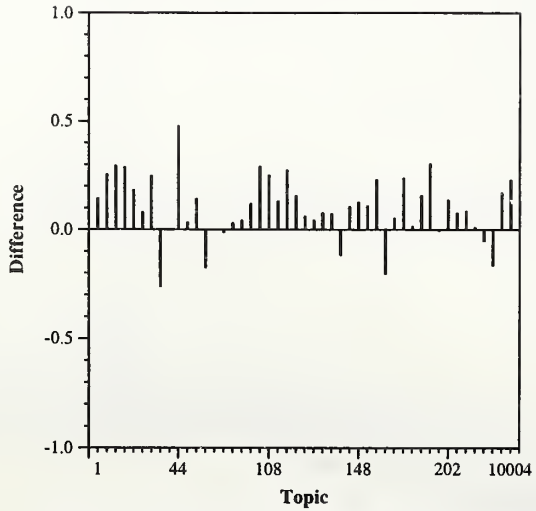
Summary Statistics	
Run Number	city6r2
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5225

Recall Level Precision Averages	
Recall	Precision
0.00	0.8510
0.10	0.7073
0.20	0.6161
0.30	0.5091
0.40	0.4400
0.50	0.3740
0.60	0.3167
0.70	0.2482
0.80	0.1745
0.90	0.0829
1.00	0.0179
Average precision over all relevant docs	
non-interpolated	0.3784

Document Level Averages	
	Precision
At 5 docs	0.6681
At 10 docs	0.6255
At 15 docs	0.5901
At 20 docs	0.5543
At 30 docs	0.5227
At 100 docs	0.3902
At 200 docs	0.2966
At 500 docs	0.1798
At 1000 docs	0.1112
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3991



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	CLCOMB
Run Description	Category A
Number of Topics	47

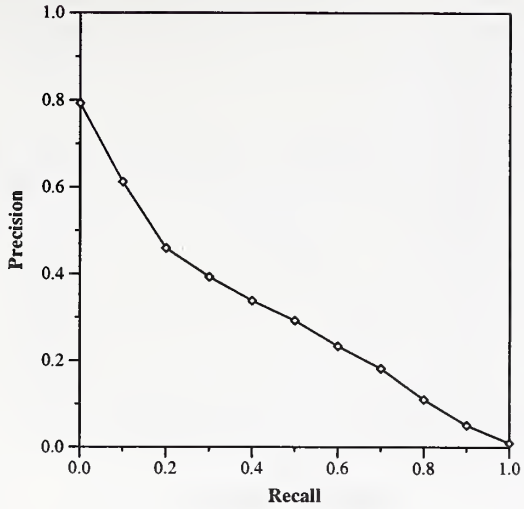
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4994

Recall Level Precision Averages	
Recall	Precision
0.00	0.7927
0.10	0.6121
0.20	0.4595
0.30	0.3931
0.40	0.3384
0.50	0.2925
0.60	0.2336
0.70	0.1818
0.80	0.1104
0.90	0.0508
1.00	0.0098

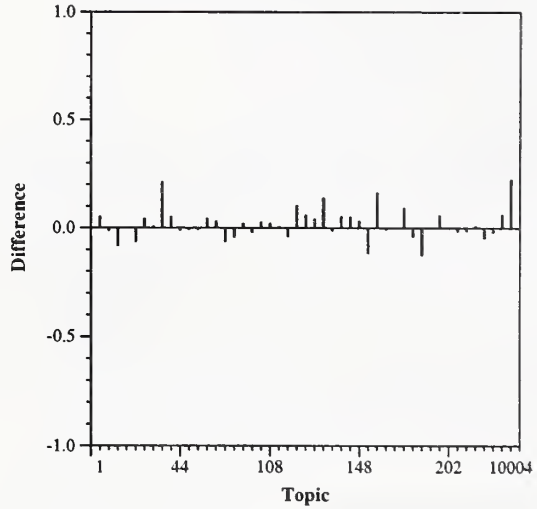
Average precision over all relevant docs	
non-interpolated	0.2961

Document Level Averages	
	Precision
At 5 docs	0.5532
At 10 docs	0.5149
At 15 docs	0.4780
At 20 docs	0.4585
At 30 docs	0.4305
At 100 docs	0.3315
At 200 docs	0.2587
At 500 docs	0.1642
At 1000 docs	0.1063

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3334



Recall-Precision Curve

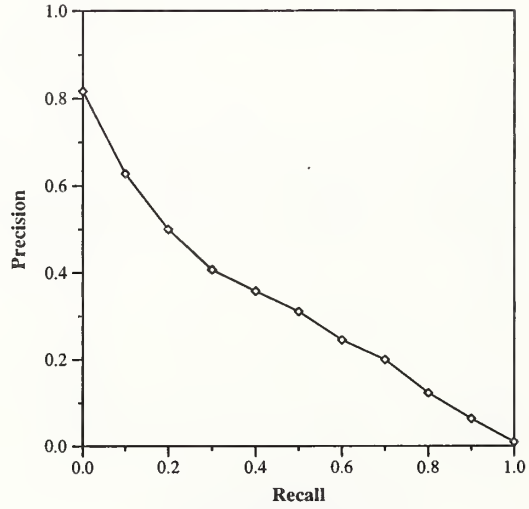


Difference from Median in Average Precision per Topic

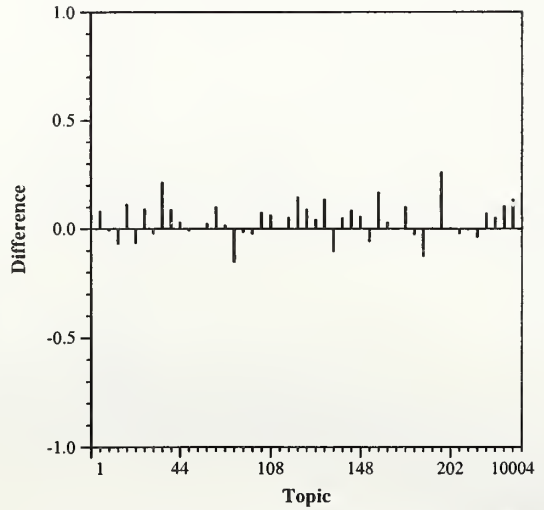
Summary Statistics	
Run Number	CLMAX
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5041

Recall Level Precision Averages	
Recall	Precision
0.00	0.8170
0.10	0.6281
0.20	0.5000
0.30	0.4070
0.40	0.3579
0.50	0.3111
0.60	0.2452
0.70	0.1998
0.80	0.1230
0.90	0.0635
1.00	0.0095
Average precision over all relevant docs	
non-interpolated	0.3146

Document Level Averages	
	Precision
At 5 docs	0.5745
At 10 docs	0.5255
At 15 docs	0.4780
At 20 docs	0.4681
At 30 docs	0.4504
At 100 docs	0.3487
At 200 docs	0.2663
At 500 docs	0.1666
At 1000 docs	0.1073
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3536



Recall-Precision Curve

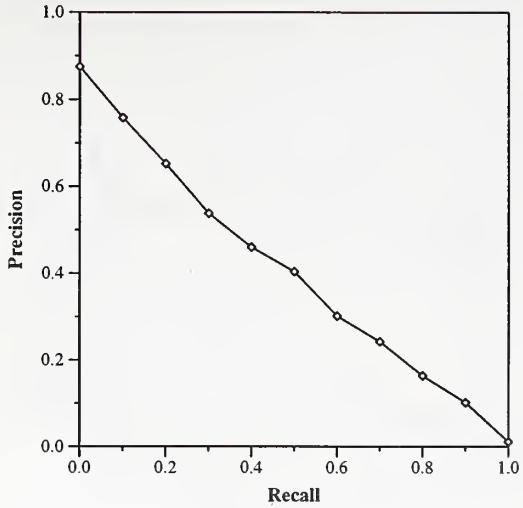


Difference from Median in Average Precision per Topic

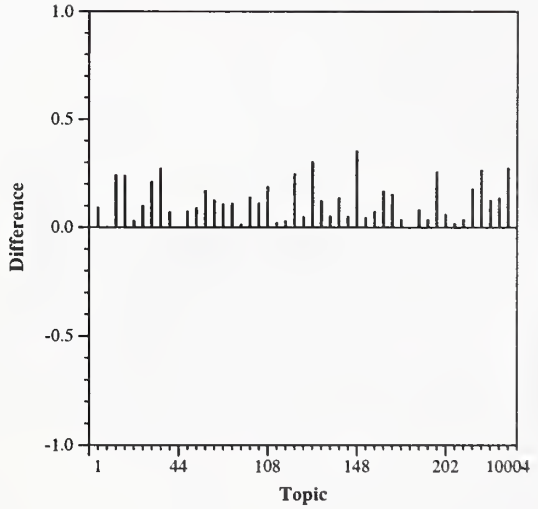
Summary Statistics	
Run Number	Cor6R1cc
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5429

Recall Level Precision Averages	
Recall	Precision
0.00	0.8754
0.10	0.7584
0.20	0.6528
0.30	0.5381
0.40	0.4604
0.50	0.4042
0.60	0.3018
0.70	0.2429
0.80	0.1639
0.90	0.1022
1.00	0.0113
Average precision over all relevant docs	
non-interpolated	0.3983

Document Level Averages	
	Precision
At 5 docs	0.6979
At 10 docs	0.6426
At 15 docs	0.6014
At 20 docs	0.5660
At 30 docs	0.5326
At 100 docs	0.3930
At 200 docs	0.3096
At 500 docs	0.1869
At 1000 docs	0.1155
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4198



Recall-Precision Curve

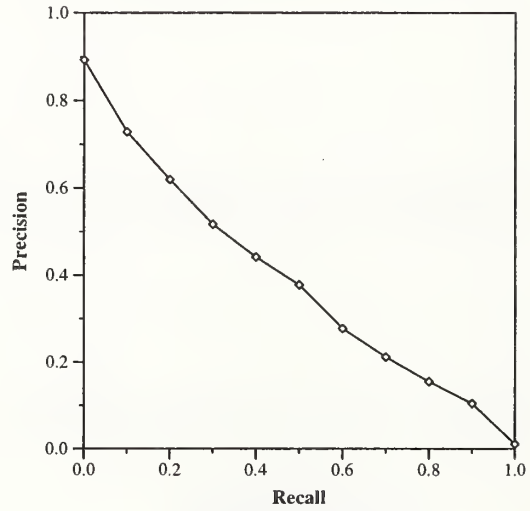


Difference from Median in Average Precision per Topic

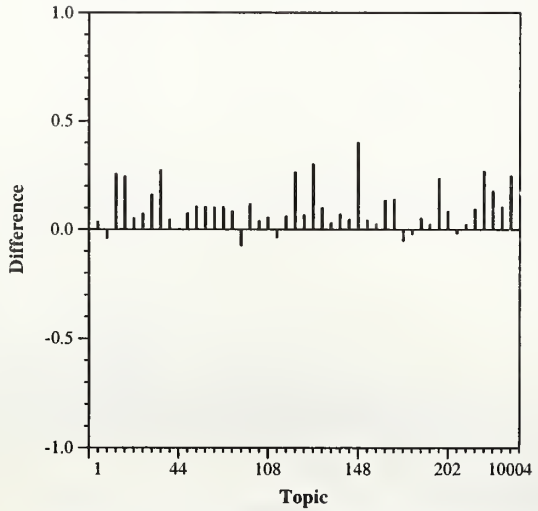
Summary Statistics	
Run Number	Cor6R2qtc
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5233

Recall Level Precision Averages	
Recall	Precision
0.00	0.8930
0.10	0.7288
0.20	0.6196
0.30	0.5169
0.40	0.4418
0.50	0.3780
0.60	0.2777
0.70	0.2119
0.80	0.1551
0.90	0.1047
1.00	0.0109
Average precision over all relevant docs	
non-interpolated	0.3766

Document Level Averages	
	Precision
At 5 docs	0.6468
At 10 docs	0.6234
At 15 docs	0.5915
At 20 docs	0.5596
At 30 docs	0.5156
At 100 docs	0.3802
At 200 docs	0.2948
At 500 docs	0.1788
At 1000 docs	0.1113
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3999



Recall-Precision Curve



Difference from Median in Average Precision per Topic

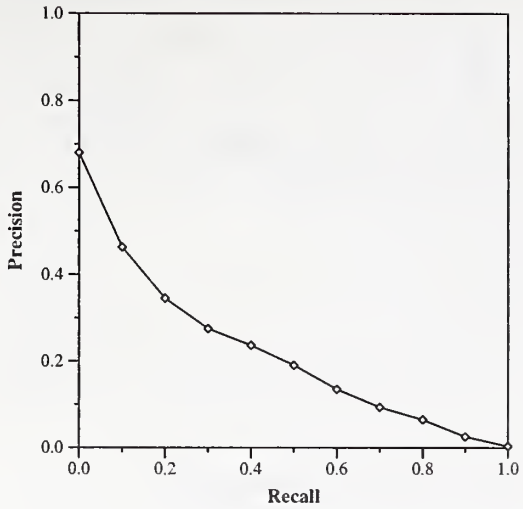
Summary Statistics	
Run Number	csiro97r1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4154

Recall Level Precision Averages	
Recall	Precision
0.00	0.6805
0.10	0.4632
0.20	0.3455
0.30	0.2755
0.40	0.2369
0.50	0.1910
0.60	0.1355
0.70	0.0939
0.80	0.0649
0.90	0.0258
1.00	0.0033

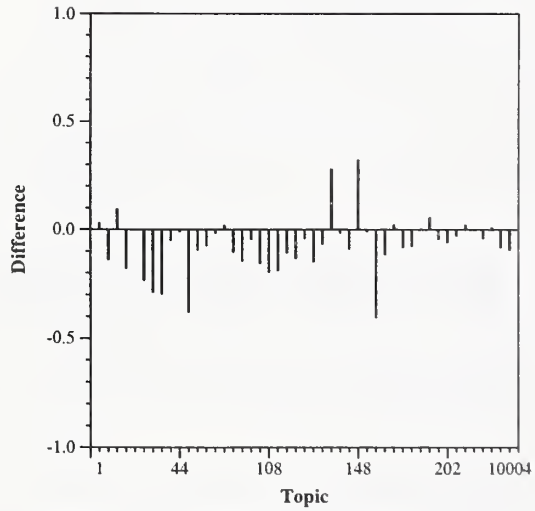
Average precision over all relevant docs	
non-interpolated	0.2068

Document Level Averages	
	Precision
At 5 docs	0.4894
At 10 docs	0.4468
At 15 docs	0.4227
At 20 docs	0.3957
At 30 docs	0.3638
At 100 docs	0.2728
At 200 docs	0.2080
At 500 docs	0.1314
At 1000 docs	0.0884

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2552



Recall-Precision Curve

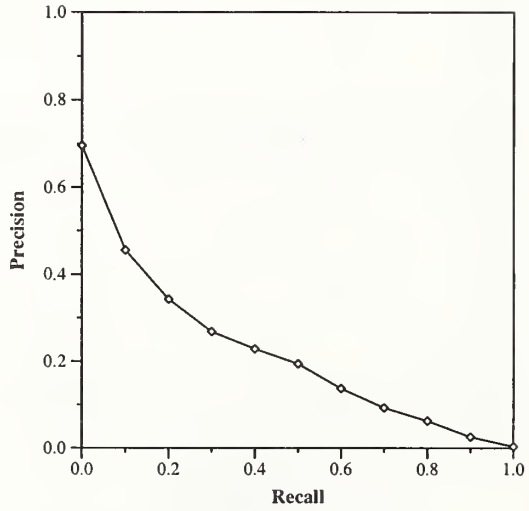


Difference from Median in Average Precision per Topic

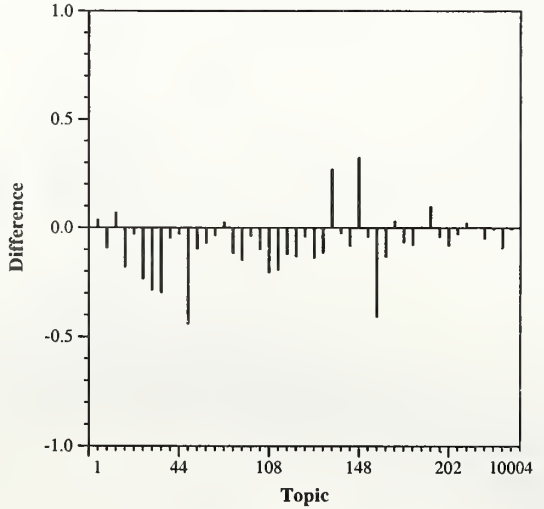
Summary Statistics	
Run Number	csiro97r2
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4200

Recall Level Precision Averages	
Recall	Precision
0.00	0.6958
0.10	0.4559
0.20	0.3433
0.30	0.2686
0.40	0.2288
0.50	0.1944
0.60	0.1375
0.70	0.0930
0.80	0.0627
0.90	0.0258
1.00	0.0035
Average precision over all relevant docs	
non-interpolated	0.2053

Document Level Averages	
	Precision
At 5 docs	0.4936
At 10 docs	0.4468
At 15 docs	0.4043
At 20 docs	0.3872
At 30 docs	0.3539
At 100 docs	0.2674
At 200 docs	0.2078
At 500 docs	0.1312
At 1000 docs	0.0894
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2503



Recall-Precision Curve

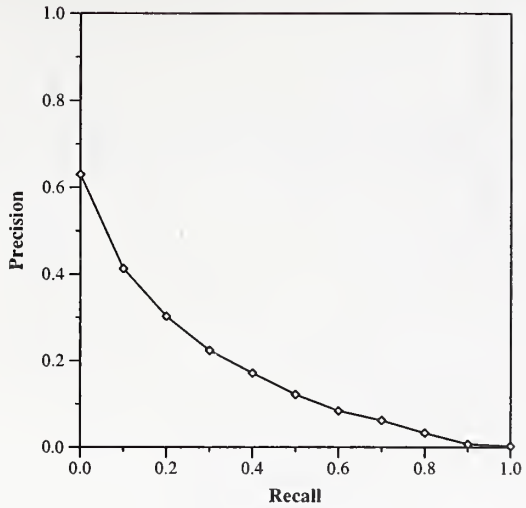


Difference from Median in Average Precision per Topic

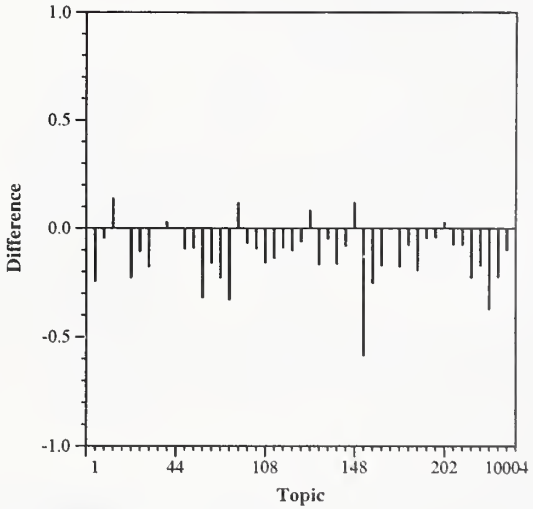
Summary Statistics	
Run Number	dbulm1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	3334

Recall Level Precision Averages	
Recall	Precision
0.00	0.6299
0.10	0.4131
0.20	0.3027
0.30	0.2240
0.40	0.1718
0.50	0.1222
0.60	0.0845
0.70	0.0620
0.80	0.0330
0.90	0.0073
1.00	0.0029
Average precision over all relevant docs	
non-interpolated	0.1619

Document Level Averages	
	Precision
At 5 docs	0.4213
At 10 docs	0.3830
At 15 docs	0.3645
At 20 docs	0.3553
At 30 docs	0.3326
At 100 docs	0.2436
At 200 docs	0.1820
At 500 docs	0.1102
At 1000 docs	0.0709
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2199



Recall-Precision Curve

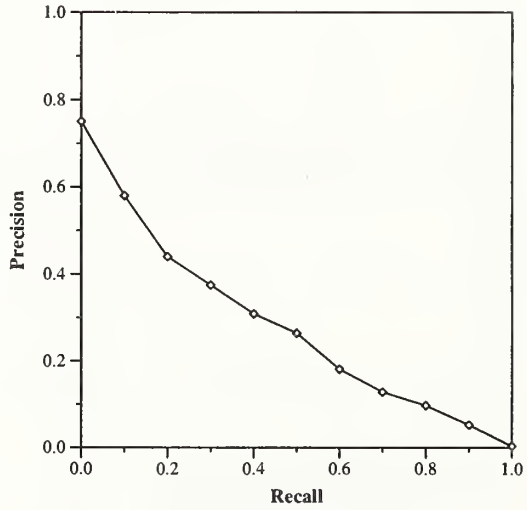


Difference from Median in Average Precision per Topic

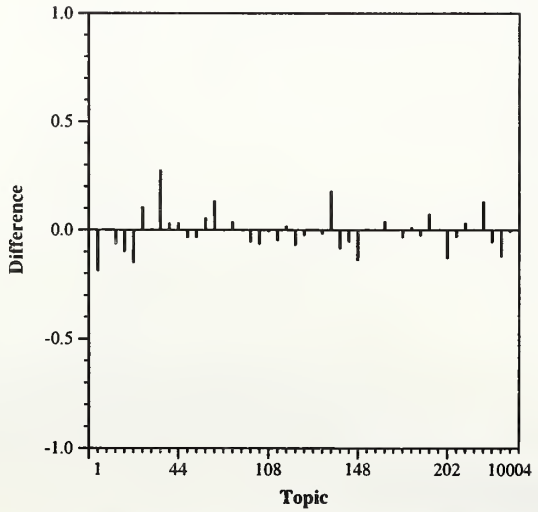
Summary Statistics	
Run Number	geroul
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4681

Recall Level Precision Averages	
Recall	Precision
0.00	0.7508
0.10	0.5802
0.20	0.4404
0.30	0.3753
0.40	0.3090
0.50	0.2645
0.60	0.1814
0.70	0.1287
0.80	0.0975
0.90	0.0525
1.00	0.0035
Average precision over all relevant docs	
non-interpolated	0.2702

Document Level Averages	
	Precision
At 5 docs	0.5532
At 10 docs	0.4787
At 15 docs	0.4468
At 20 docs	0.4170
At 30 docs	0.3950
At 100 docs	0.3102
At 200 docs	0.2434
At 500 docs	0.1557
At 1000 docs	0.0996
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3176



Recall-Precision Curve

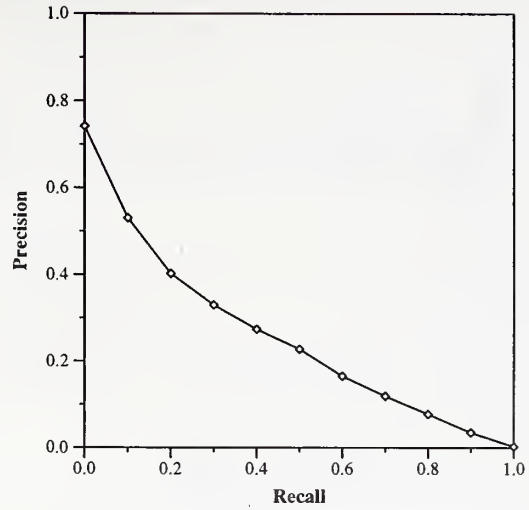


Difference from Median in Average Precision per Topic

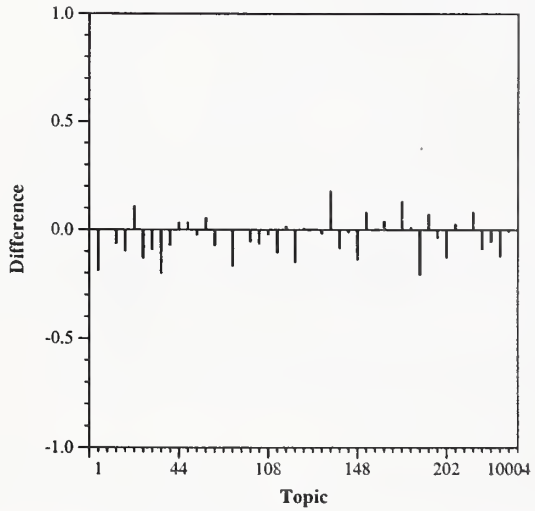
Summary Statistics	
Run Number	gesri2
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4334

Recall Level Precision Averages	
Recall	Precision
0.00	0.7418
0.10	0.5305
0.20	0.4025
0.30	0.3301
0.40	0.2737
0.50	0.2279
0.60	0.1656
0.70	0.1195
0.80	0.0778
0.90	0.0352
1.00	0.0026
Average precision over all relevant docs	
non-interpolated	0.2458

Document Level Averages	
	Precision
At 5 docs	0.5447
At 10 docs	0.4894
At 15 docs	0.4440
At 20 docs	0.4245
At 30 docs	0.3965
At 100 docs	0.2983
At 200 docs	0.2288
At 500 docs	0.1436
At 1000 docs	0.0922
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2906



Recall-Precision Curve



Difference from Median in Average Precision per Topic

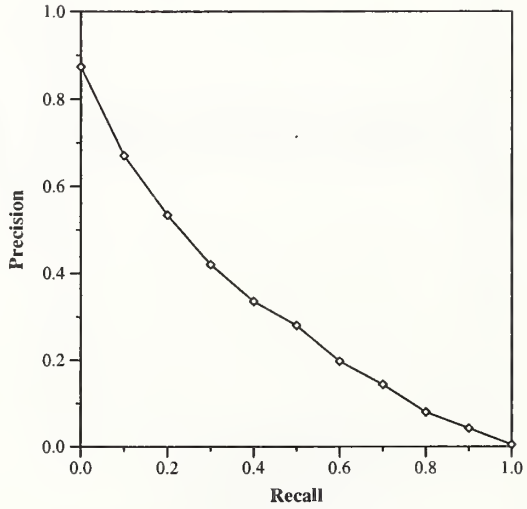
Summary Statistics	
Run Number	Mercure4
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4774

Recall Level Precision Averages	
Recall	Precision
0.00	0.8740
0.10	0.6708
0.20	0.5344
0.30	0.4207
0.40	0.3358
0.50	0.2803
0.60	0.1976
0.70	0.1439
0.80	0.0802
0.90	0.0428
1.00	0.0050

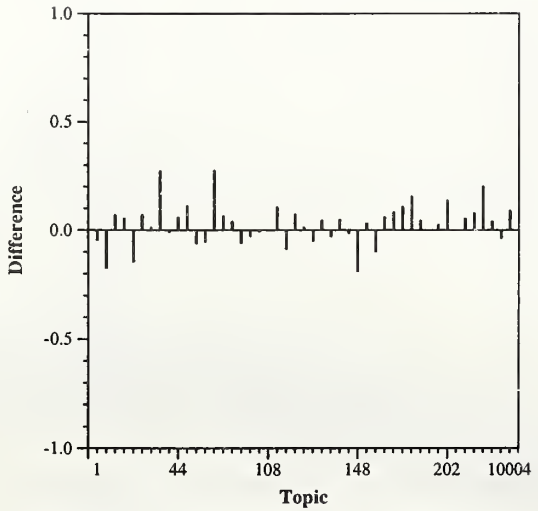
Average precision over all relevant docs	
non-interpolated	0.3061

Document Level Averages	
	Precision
At 5 docs	0.6298
At 10 docs	0.5702
At 15 docs	0.5418
At 20 docs	0.5266
At 30 docs	0.4766
At 100 docs	0.3257
At 200 docs	0.2595
At 500 docs	0.1640
At 1000 docs	0.1016

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3400



Recall-Precision Curve

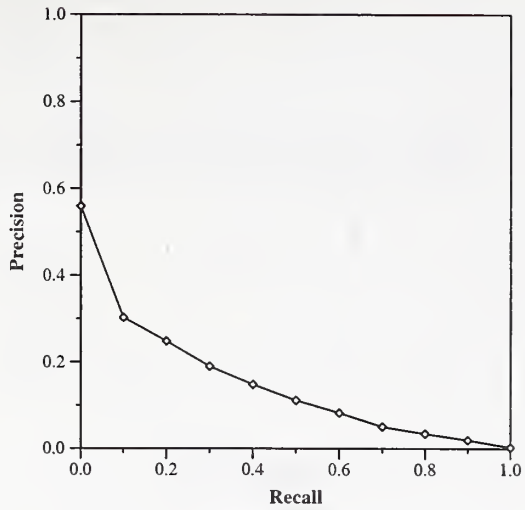


Difference from Median in Average Precision per Topic

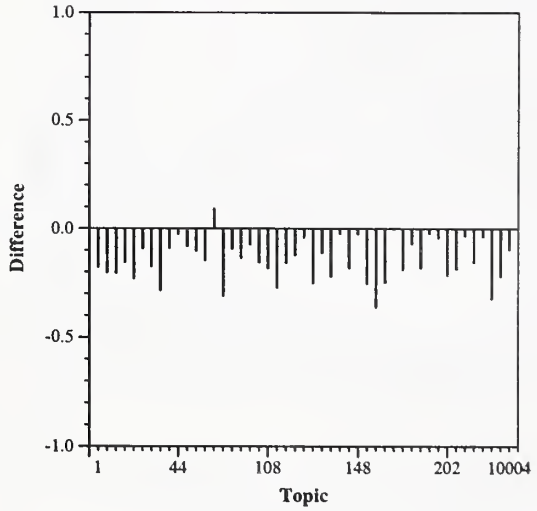
Summary Statistics	
Run Number	virtue3
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	3289

Recall Level Precision Averages	
Recall	Precision
0.00	0.5595
0.10	0.3028
0.20	0.2485
0.30	0.1901
0.40	0.1483
0.50	0.1117
0.60	0.0828
0.70	0.0507
0.80	0.0345
0.90	0.0200
1.00	0.0029
Average precision over all relevant docs	
non-interpolated	0.1318

Document Level Averages	
	Precision
At 5 docs	0.3574
At 10 docs	0.3149
At 15 docs	0.2936
At 20 docs	0.2840
At 30 docs	0.2610
At 100 docs	0.2128
At 200 docs	0.1803
At 500 docs	0.1103
At 1000 docs	0.0700
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1823



Recall-Precision Curve

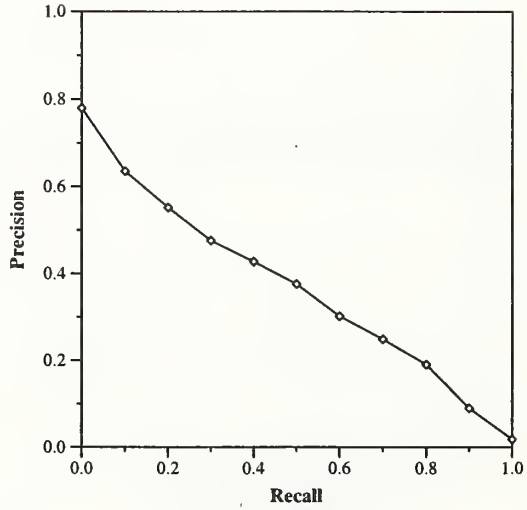


Difference from Median in Average Precision per Topic

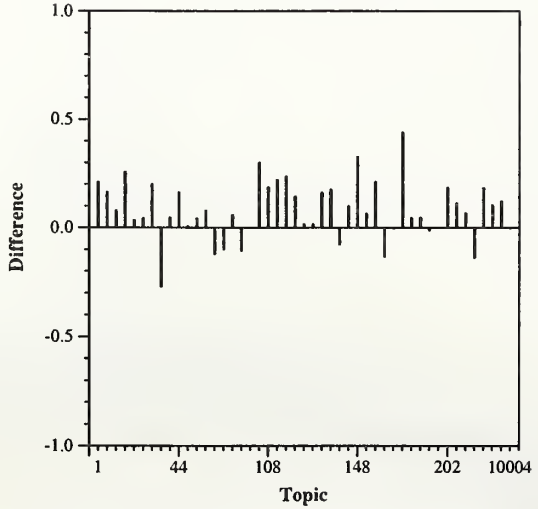
Summary Statistics	
Run Number	pirc7R1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5284

Recall Level Precision Averages	
Recall	Precision
0.00	0.7797
0.10	0.6354
0.20	0.5521
0.30	0.4761
0.40	0.4277
0.50	0.3761
0.60	0.3021
0.70	0.2492
0.80	0.1910
0.90	0.0902
1.00	0.0184
Average precision over all relevant docs	
non-interpolated	0.3605

Document Level Averages	
	Precision
At 5 docs	0.6255
At 10 docs	0.5915
At 15 docs	0.5674
At 20 docs	0.5511
At 30 docs	0.5128
At 100 docs	0.3915
At 200 docs	0.2952
At 500 docs	0.1829
At 1000 docs	0.1124
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3895



Recall-Precision Curve

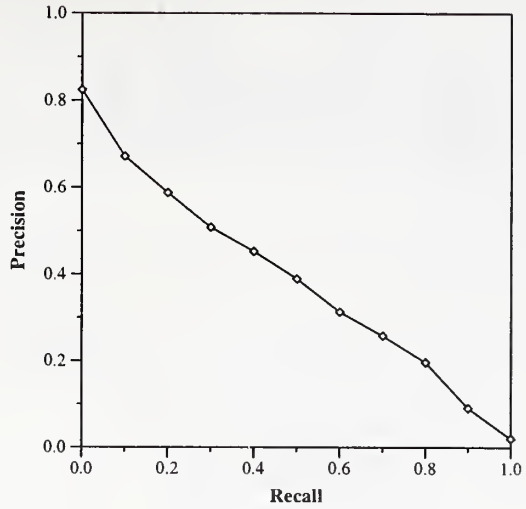


Difference from Median in Average Precision per Topic

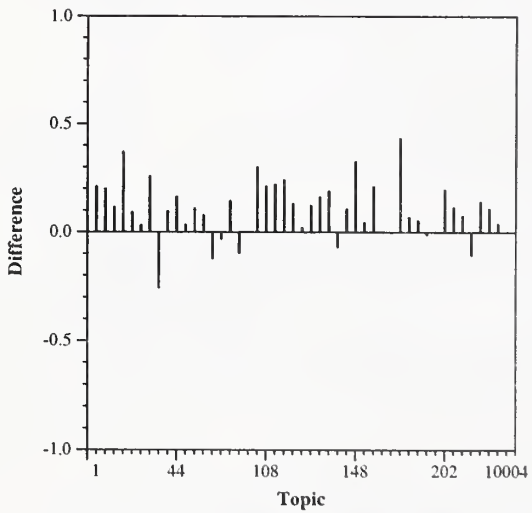
Summary Statistics	
Run Number	pirc7R2
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	5288

Recall Level Precision Averages	
Recall	Precision
0.00	0.8245
0.10	0.6716
0.20	0.5878
0.30	0.5078
0.40	0.4527
0.50	0.3896
0.60	0.3130
0.70	0.2582
0.80	0.1972
0.90	0.0914
1.00	0.0204
Average precision over all relevant docs	
non-interpolated	0.3783

Document Level Averages	
	Precision
At 5 docs	0.6511
At 10 docs	0.6191
At 15 docs	0.5915
At 20 docs	0.5755
At 30 docs	0.5312
At 100 docs	0.3962
At 200 docs	0.2999
At 500 docs	0.1839
At 1000 docs	0.1125
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4033



Recall-Precision Curve

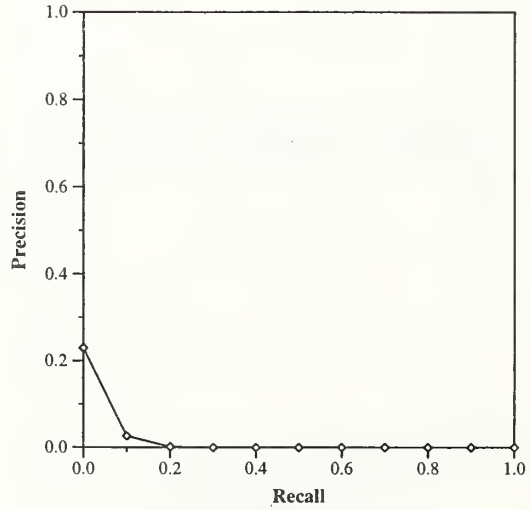


Difference from Median in Average Precision per Topic

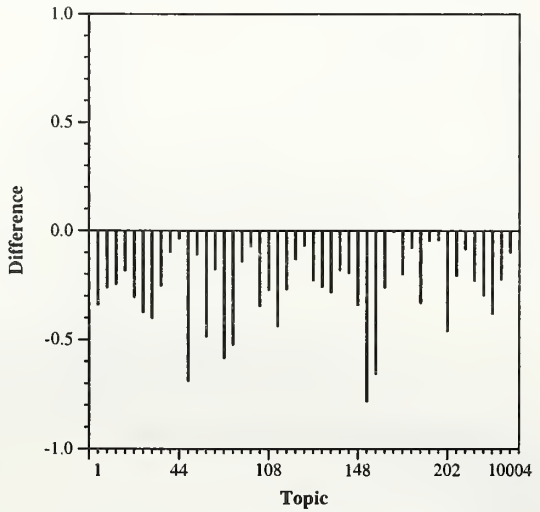
Summary Statistics	
Run Number	rutLADc1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	18562
Relevant:	6872
Rel-ret:	377

Recall Level Precision Averages	
Recall	Precision
0.00	0.2297
0.10	0.0263
0.20	0.0016
0.30	0.0000
0.40	0.0000
0.50	0.0000
0.60	0.0000
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.0087

Document Level Averages	
	Precision
At 5 docs	0.1191
At 10 docs	0.0957
At 15 docs	0.0879
At 20 docs	0.0787
At 30 docs	0.0667
At 100 docs	0.0398
At 200 docs	0.0262
At 500 docs	0.0131
At 1000 docs	0.0080
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0275



Recall-Precision Curve



Difference from Median in Average Precision per Topic

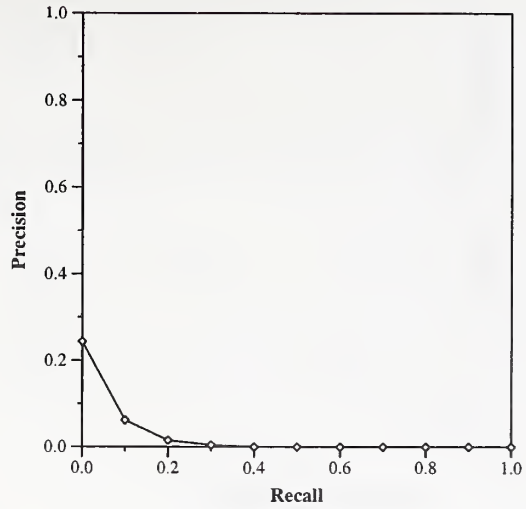
Summary Statistics	
Run Number	rutLADw1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	18562
Relevant:	6872
Rel-ret:	439

Recall Level Precision Averages	
Recall	Precision
0.00	0.2439
0.10	0.0621
0.20	0.0153
0.30	0.0043
0.40	0.0000
0.50	0.0000
0.60	0.0000
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000

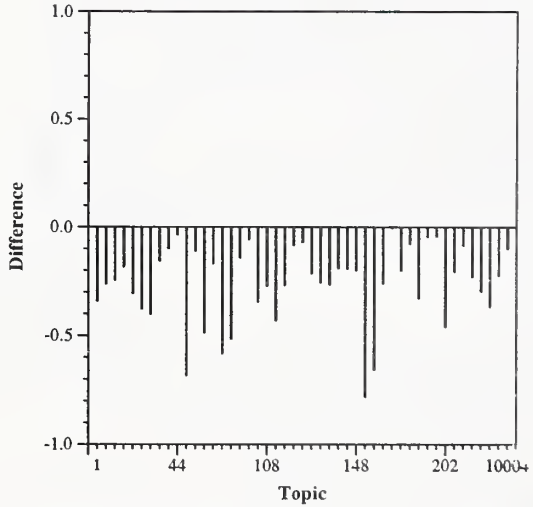
Average precision over all relevant docs	
non-interpolated	0.0163

Document Level Averages	
	Precision
At 5 docs	0.1404
At 10 docs	0.1149
At 15 docs	0.1135
At 20 docs	0.0979
At 30 docs	0.0844
At 100 docs	0.0521
At 200 docs	0.0334
At 500 docs	0.0159
At 1000 docs	0.0093

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0364



Recall-Precision Curve

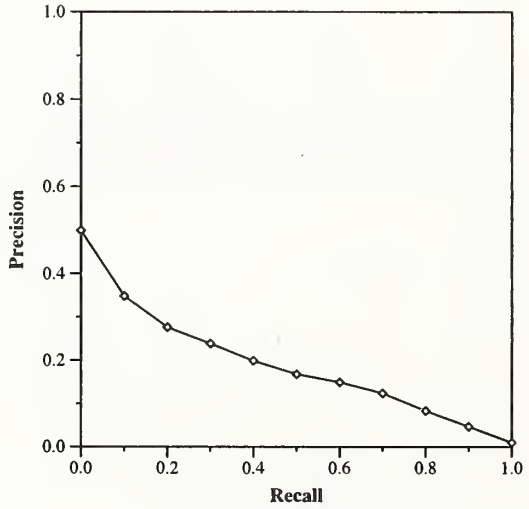


Difference from Median in Average Precision per Topic

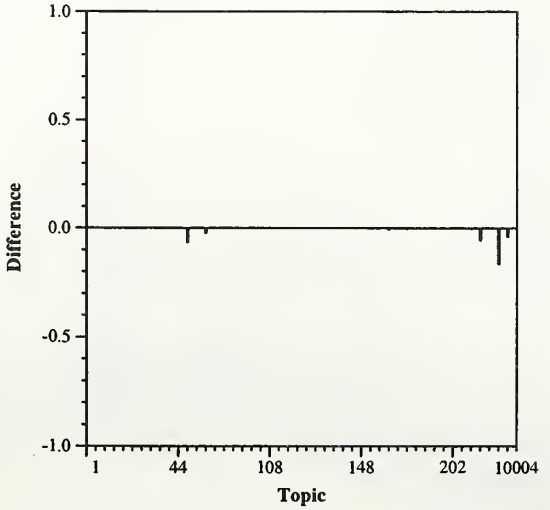
Summary Statistics	
Run Number	teklis
Run Description	Category B
Number of Topics	46
Total number of documents over all topics	
Retrieved:	46000
Relevant:	3499
Rel-ret:	2580

Recall Level Precision Averages	
Recall	Precision
0.00	0.4988
0.10	0.3484
0.20	0.2765
0.30	0.2388
0.40	0.1990
0.50	0.1682
0.60	0.1497
0.70	0.1243
0.80	0.0837
0.90	0.0472
1.00	0.0101
Average precision over all relevant docs	
non-interpolated	0.1774

Document Level Averages	
	Precision
At 5 docs	0.3000
At 10 docs	0.3022
At 15 docs	0.2884
At 20 docs	0.2728
At 30 docs	0.2652
At 100 docs	0.2115
At 200 docs	0.1577
At 500 docs	0.0920
At 1000 docs	0.0561
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2053



Recall-Precision Curve

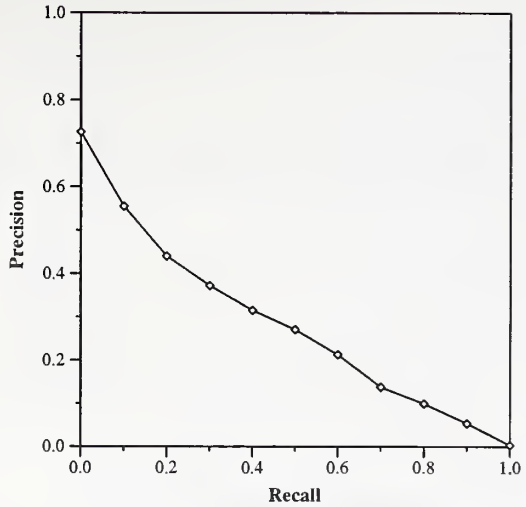


Difference from Median in Average Precision per Topic

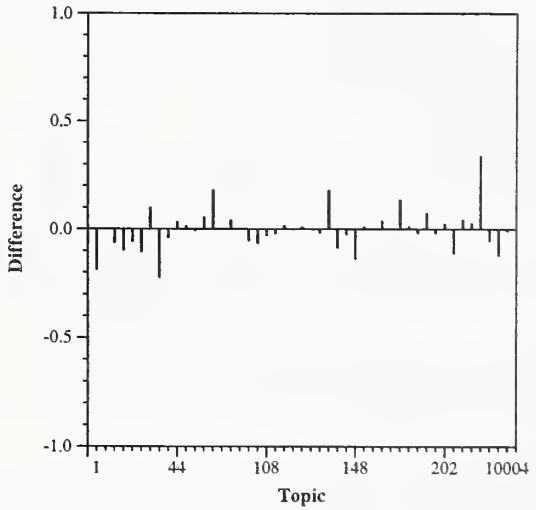
Summary Statistics	
Run Number	srige1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4878

Recall Level Precision Averages	
Recall	Precision
0.00	0.7264
0.10	0.5553
0.20	0.4407
0.30	0.3726
0.40	0.3155
0.50	0.2709
0.60	0.2131
0.70	0.1380
0.80	0.0996
0.90	0.0541
1.00	0.0035
Average precision over all relevant docs	
non-interpolated	0.2730

Document Level Averages	
	Precision
At 5 docs	0.5574
At 10 docs	0.5021
At 15 docs	0.4794
At 20 docs	0.4564
At 30 docs	0.4206
At 100 docs	0.3087
At 200 docs	0.2429
At 500 docs	0.1598
At 1000 docs	0.1038
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3126



Recall-Precision Curve

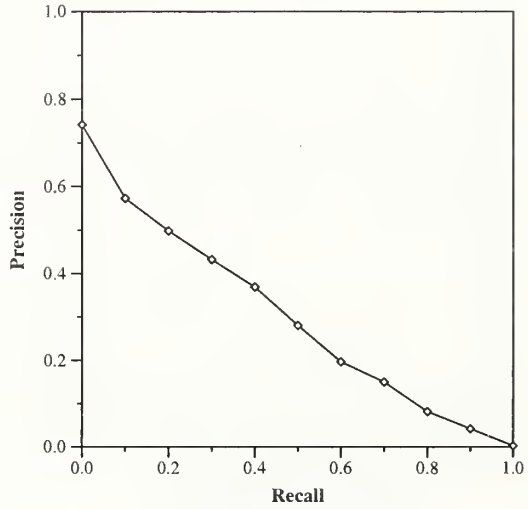


Difference from Median in Average Precision per Topic

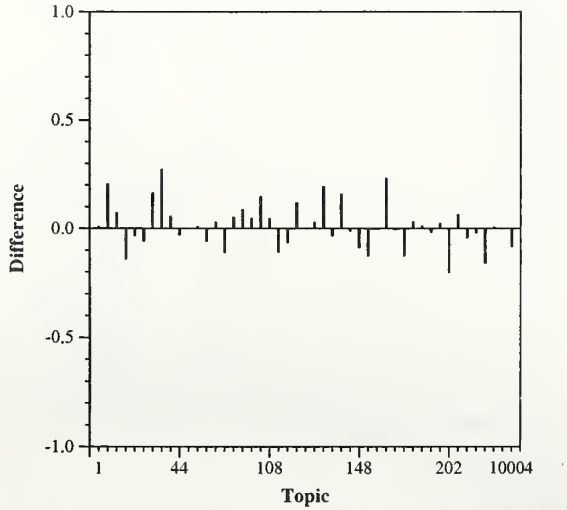
Summary Statistics	
Run Number	ETH6R1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4653

Recall Level Precision Averages	
Recall	Precision
0.00	0.7418
0.10	0.5729
0.20	0.4981
0.30	0.4325
0.40	0.3691
0.50	0.2806
0.60	0.1968
0.70	0.1500
0.80	0.0819
0.90	0.0421
1.00	0.0030
Average precision over all relevant docs	
non-interpolated	0.2894

Document Level Averages	
	Precision
At 5 docs	0.5830
At 10 docs	0.5489
At 15 docs	0.5177
At 20 docs	0.5000
At 30 docs	0.4539
At 100 docs	0.3417
At 200 docs	0.2614
At 500 docs	0.1615
At 1000 docs	0.0990
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3372



Recall-Precision Curve



Difference from Median in Average Precision per Topic

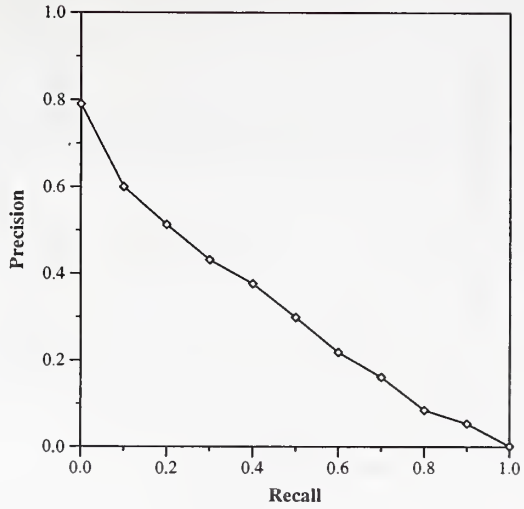
Summary Statistics	
Run Number	ETH6R2
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4925

Recall Level Precision Averages	
Recall	Precision
0.00	0.7903
0.10	0.6001
0.20	0.5132
0.30	0.4321
0.40	0.3771
0.50	0.2995
0.60	0.2189
0.70	0.1616
0.80	0.0851
0.90	0.0537
1.00	0.0016

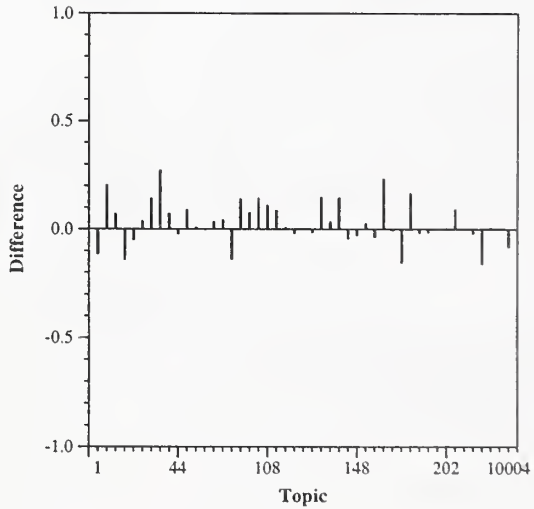
Average precision over all relevant docs	
non-interpolated	0.3059

Document Level Averages	
	Precision
At 5 docs	0.6298
At 10 docs	0.5851
At 15 docs	0.5248
At 20 docs	0.5106
At 30 docs	0.4645
At 100 docs	0.3509
At 200 docs	0.2698
At 500 docs	0.1699
At 1000 docs	0.1048

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3361



Recall-Precision Curve

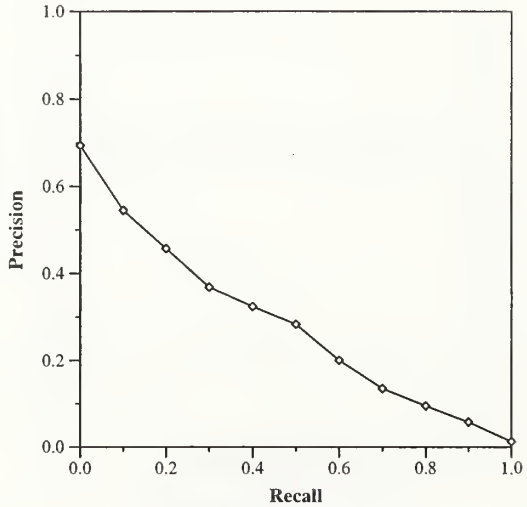


Difference from Median in Average Precision per Topic

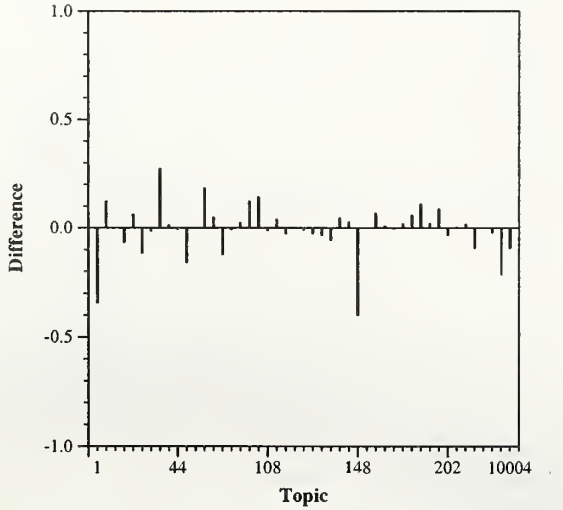
Summary Statistics	
Run Number	Brkly19
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4778

Recall Level Precision Averages	
Recall	Precision
0.00	0.6941
0.10	0.5453
0.20	0.4576
0.30	0.3688
0.40	0.3242
0.50	0.2833
0.60	0.2006
0.70	0.1354
0.80	0.0951
0.90	0.0580
1.00	0.0130
Average precision over all relevant docs	
non-interpolated	0.2705

Document Level Averages	
	Precision
At 5 docs	0.4809
At 10 docs	0.4426
At 15 docs	0.4270
At 20 docs	0.4213
At 30 docs	0.4007
At 100 docs	0.3270
At 200 docs	0.2530
At 500 docs	0.1621
At 1000 docs	0.1017
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3259



Recall-Precision Curve

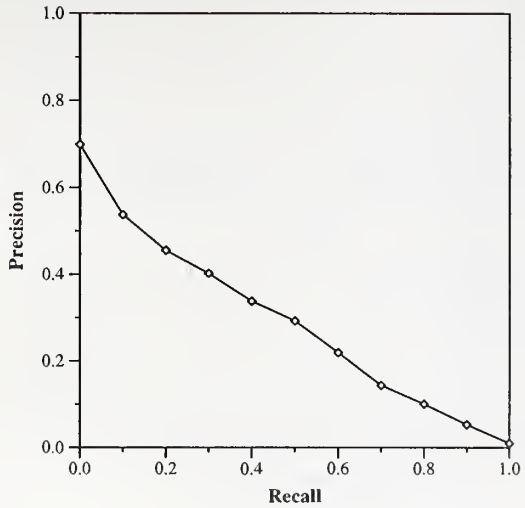


Difference from Median in Average Precision per Topic

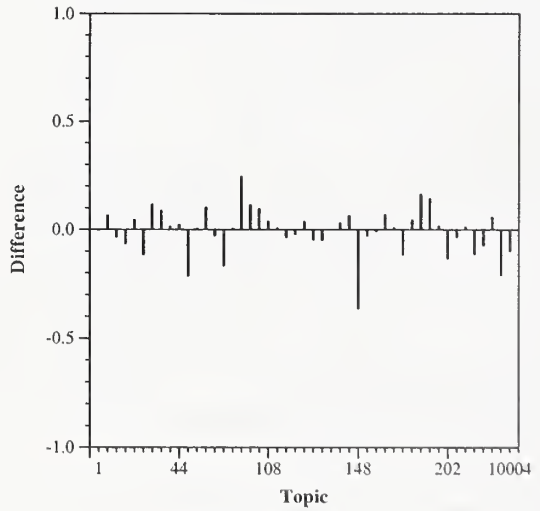
Summary Statistics	
Run Number	Brkly20
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4903

Recall Level Precision Averages	
Recall	Precision
0.00	0.6991
0.10	0.5377
0.20	0.4557
0.30	0.4026
0.40	0.3386
0.50	0.2930
0.60	0.2199
0.70	0.1443
0.80	0.1009
0.90	0.0534
1.00	0.0102
Average precision over all relevant docs	
non-interpolated	0.2709

Document Level Averages	
	Precision
At 5 docs	0.4681
At 10 docs	0.4340
At 15 docs	0.4156
At 20 docs	0.4160
At 30 docs	0.4050
At 100 docs	0.3323
At 200 docs	0.2610
At 500 docs	0.1662
At 1000 docs	0.1043
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3317



Recall-Precision Curve

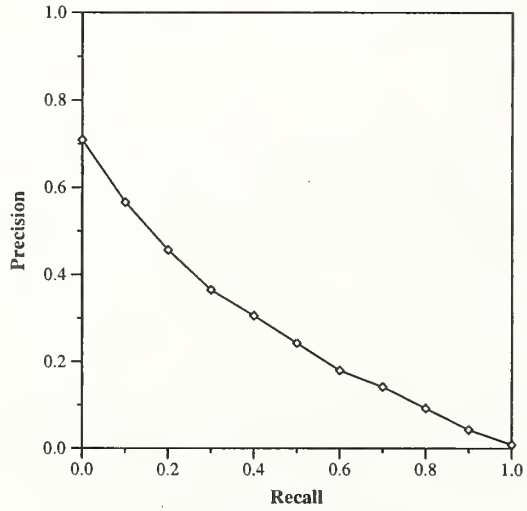


Difference from Median in Average Precision per Topic

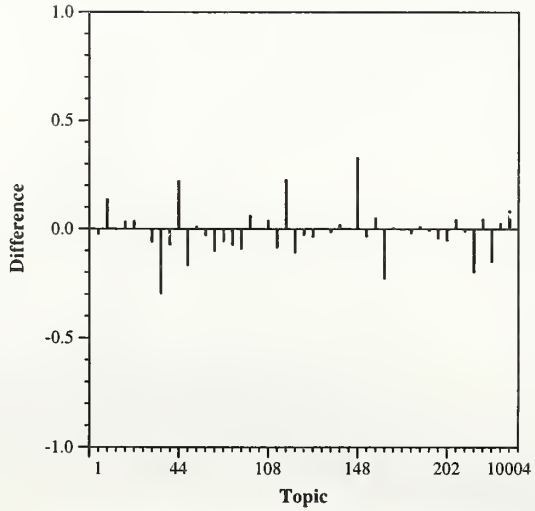
Summary Statistics	
Run Number	UCSDrt6
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4642

Recall Level Precision Averages	
Recall	Precision
0.00	0.7093
0.10	0.5666
0.20	0.4571
0.30	0.3654
0.40	0.3060
0.50	0.2428
0.60	0.1798
0.70	0.1420
0.80	0.0926
0.90	0.0432
1.00	0.0088
Average precision over all relevant docs	
non-interpolated	0.2650

Document Level Averages	
	Precision
At 5 docs	0.5319
At 10 docs	0.5128
At 15 docs	0.4879
At 20 docs	0.4617
At 30 docs	0.4305
At 100 docs	0.3294
At 200 docs	0.2541
At 500 docs	0.1548
At 1000 docs	0.0988
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3089



Recall-Precision Curve

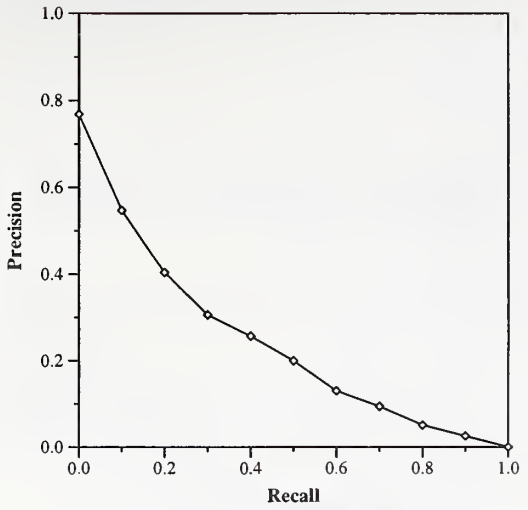


Difference from Median in Average Precision per Topic

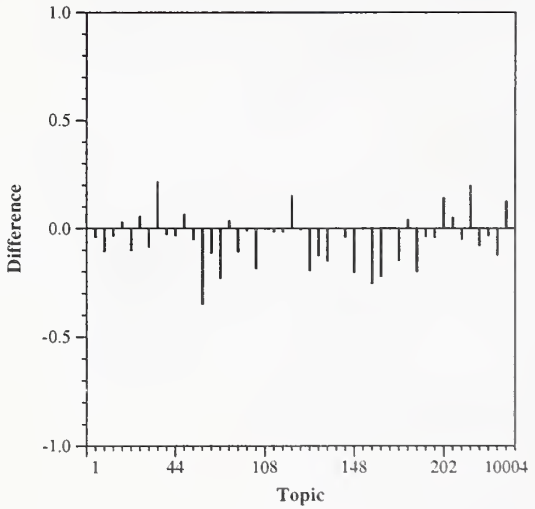
Summary Statistics	
Run Number	INQ403
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4047

Recall Level Precision Averages	
Recall	Precision
0.00	0.7686
0.10	0.5473
0.20	0.4042
0.30	0.3062
0.40	0.2567
0.50	0.1998
0.60	0.1303
0.70	0.0942
0.80	0.0511
0.90	0.0256
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.2291

Document Level Averages	
	Precision
At 5 docs	0.5532
At 10 docs	0.5170
At 15 docs	0.4936
At 20 docs	0.4585
At 30 docs	0.4227
At 100 docs	0.2874
At 200 docs	0.2212
At 500 docs	0.1352
At 1000 docs	0.0861
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2897



Recall-Precision Curve

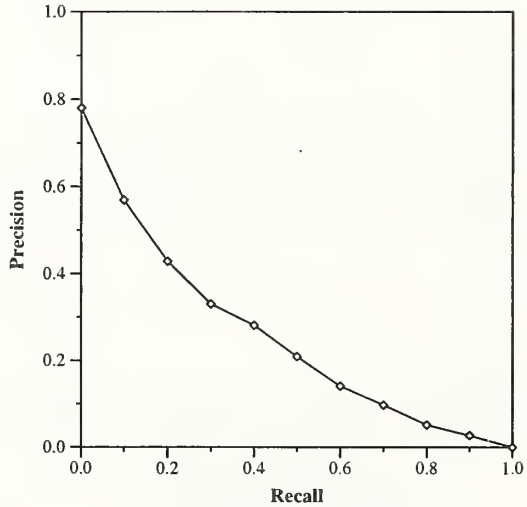


Difference from Median in Average Precision per Topic

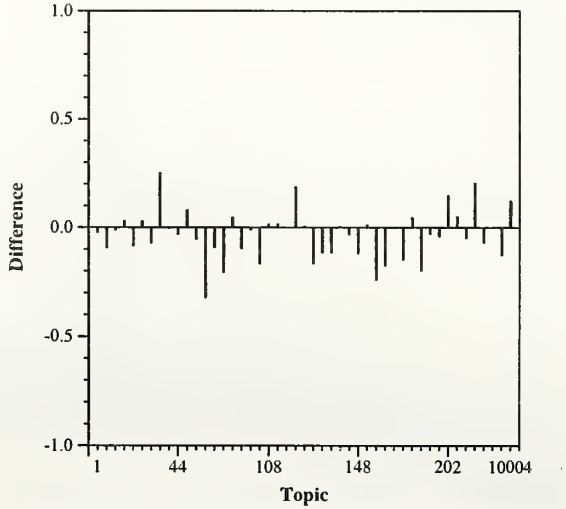
Summary Statistics	
Run Number	INQ404
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4121

Recall Level Precision Averages	
Recall	Precision
0.00	0.7804
0.10	0.5696
0.20	0.4288
0.30	0.3302
0.40	0.2814
0.50	0.2094
0.60	0.1412
0.70	0.0977
0.80	0.0516
0.90	0.0271
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.2429

Document Level Averages	
	Precision
At 5 docs	0.5532
At 10 docs	0.5362
At 15 docs	0.5078
At 20 docs	0.4755
At 30 docs	0.4376
At 100 docs	0.3009
At 200 docs	0.2312
At 500 docs	0.1398
At 1000 docs	0.0877
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3055



Recall-Precision Curve

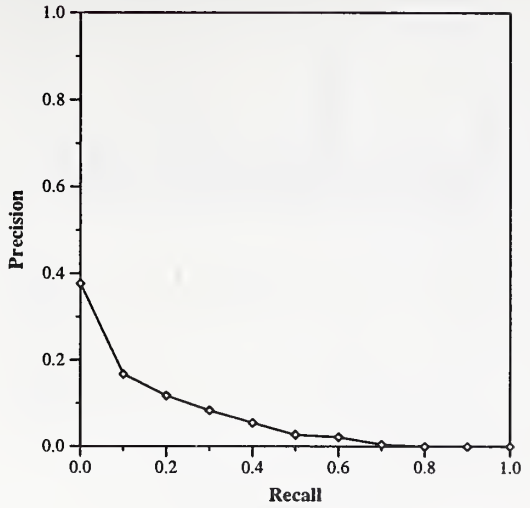


Difference from Median in Average Precision per Topic

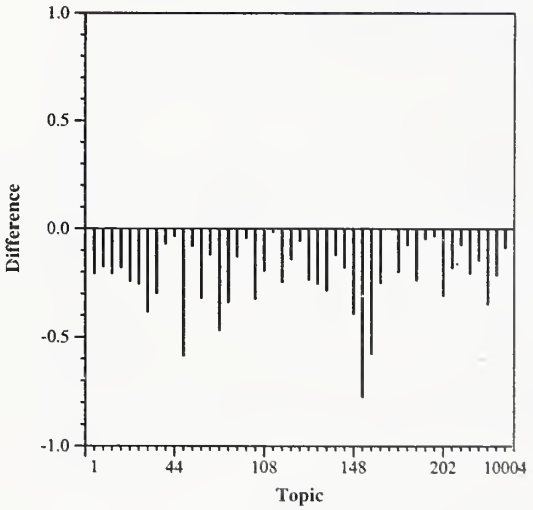
Summary Statistics	
Run Number	ispr1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	2651

Recall Level Precision Averages	
Recall	Precision
0.00	0.3769
0.10	0.1673
0.20	0.1178
0.30	0.0833
0.40	0.0548
0.50	0.0269
0.60	0.0216
0.70	0.0043
0.80	0.0000
0.90	0.0000
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.0581

Document Level Averages	
	Precision
At 5 docs	0.1745
At 10 docs	0.1787
At 15 docs	0.1631
At 20 docs	0.1681
At 30 docs	0.1596
At 100 docs	0.1406
At 200 docs	0.1137
At 500 docs	0.0784
At 1000 docs	0.0564
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1188



Recall-Precision Curve

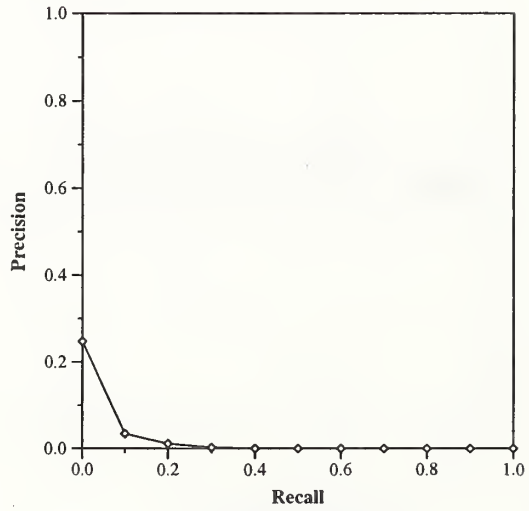


Difference from Median in Average Precision per Topic

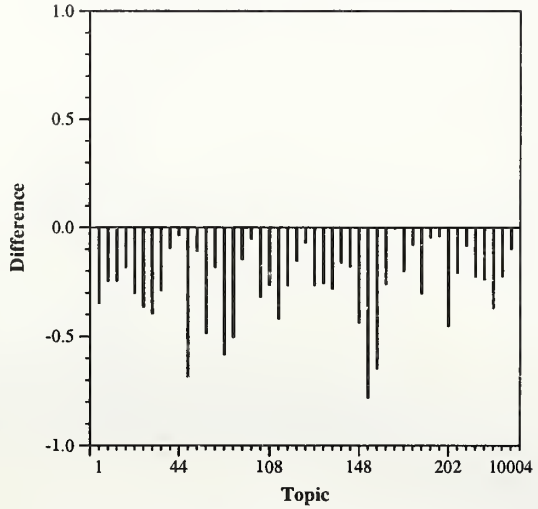
Summary Statistics	
Run Number	ispr2
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	924

Recall Level Precision Averages	
Recall	Precision
0.00	0.2475
0.10	0.0343
0.20	0.0112
0.30	0.0022
0.40	0.0000
0.50	0.0000
0.60	0.0000
0.70	0.0000
0.80	0.0000
0.90	0.0000
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.0102

Document Level Averages	
	Precision
At 5 docs	0.0809
At 10 docs	0.0702
At 15 docs	0.0681
At 20 docs	0.0702
At 30 docs	0.0688
At 100 docs	0.0515
At 200 docs	0.0396
At 500 docs	0.0281
At 1000 docs	0.0197
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0375



Recall-Precision Curve



Difference from Median in Average Precision per Topic

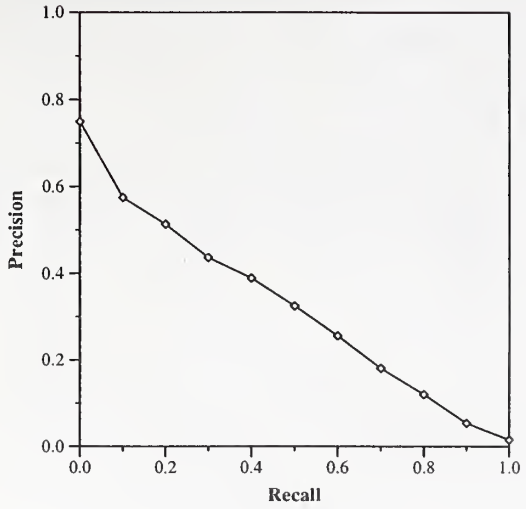
Summary Statistics	
Run Number	uwmt6r0
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	34192
Relevant:	6872
Rel-ret:	4793

Recall Level Precision Averages	
Recall	Precision
0.00	0.7498
0.10	0.5750
0.20	0.5135
0.30	0.4367
0.40	0.3895
0.50	0.3252
0.60	0.2560
0.70	0.1810
0.80	0.1207
0.90	0.0537
1.00	0.0153

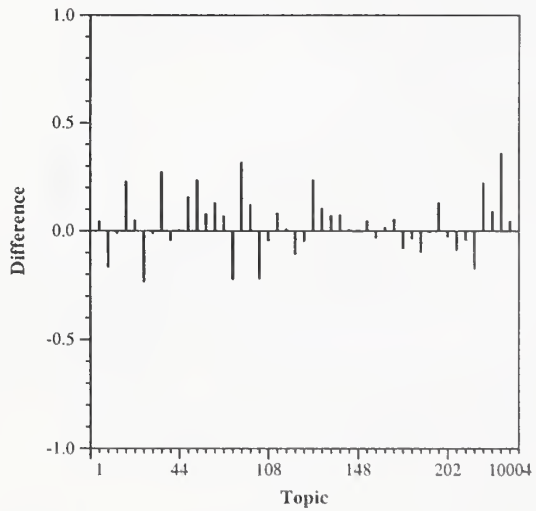
Average precision over all relevant docs	
non-interpolated	0.3111

Document Level Averages	
	Precision
At 5 docs	0.5277
At 10 docs	0.5085
At 15 docs	0.5007
At 20 docs	0.4819
At 30 docs	0.4496
At 100 docs	0.3496
At 200 docs	0.2639
At 500 docs	0.1611
At 1000 docs	0.1020

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3604



Recall-Precision Curve

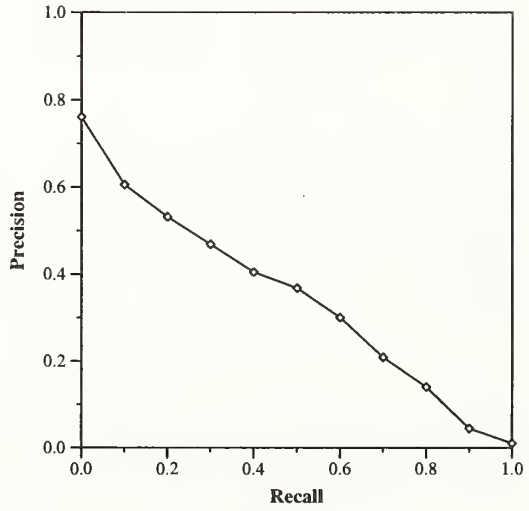


Difference from Median in Average Precision per Topic

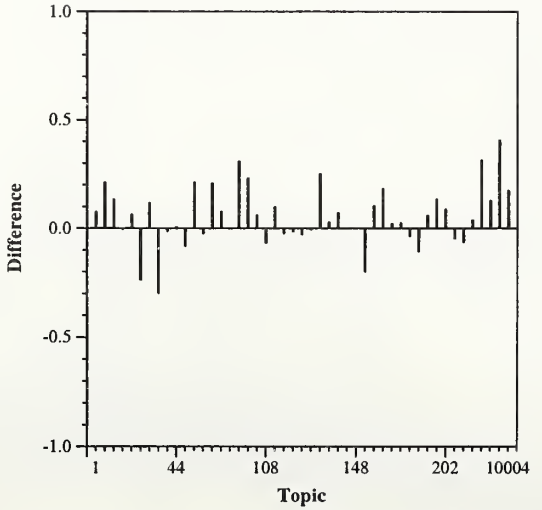
Summary Statistics	
Run Number	uwmt6r1
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	37635
Relevant:	6872
Rel-ret:	5036

Recall Level Precision Averages	
Recall	Precision
0.00	0.7612
0.10	0.6062
0.20	0.5321
0.30	0.4690
0.40	0.4052
0.50	0.3683
0.60	0.3008
0.70	0.2090
0.80	0.1405
0.90	0.0446
1.00	0.0104
Average precision over all relevant docs	
non-interpolated	0.3326

Document Level Averages	
	Precision
At 5 docs	0.5745
At 10 docs	0.5617
At 15 docs	0.5390
At 20 docs	0.5106
At 30 docs	0.4794
At 100 docs	0.3553
At 200 docs	0.2762
At 500 docs	0.1717
At 1000 docs	0.1071
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3769



Recall-Precision Curve

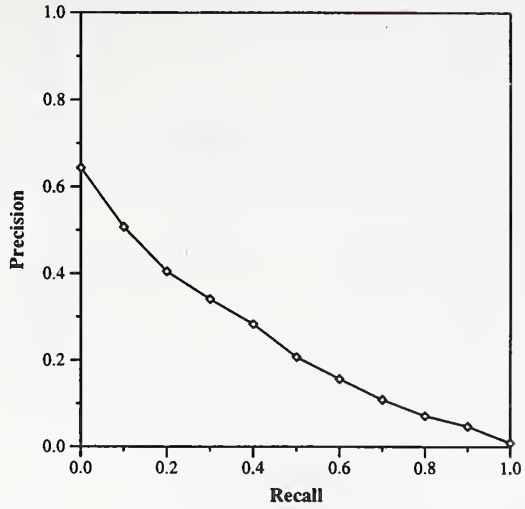


Difference from Median in Average Precision per Topic

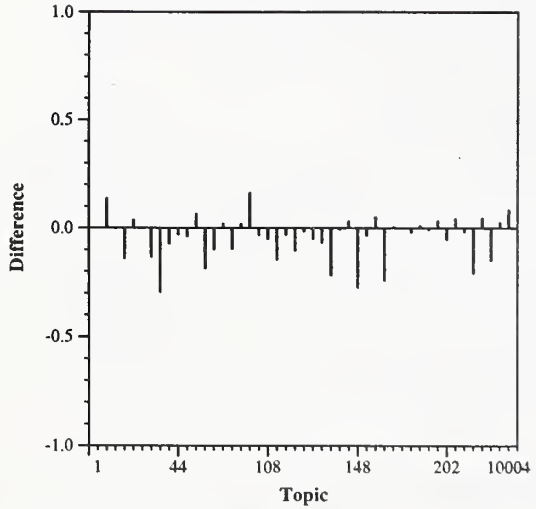
Summary Statistics	
Run Number	VrtyRT6
Run Description	Category A
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	6872
Rel-ret:	4421

Recall Level Precision Averages	
Recall	Precision
0.00	0.6438
0.10	0.5075
0.20	0.4053
0.30	0.3413
0.40	0.2836
0.50	0.2076
0.60	0.1571
0.70	0.1091
0.80	0.0717
0.90	0.0475
1.00	0.0092
Average precision over all relevant docs	
non-interpolated	0.2343

Document Level Averages	
	Precision
At 5 docs	0.4809
At 10 docs	0.4766
At 15 docs	0.4667
At 20 docs	0.4394
At 30 docs	0.4262
At 100 docs	0.3185
At 200 docs	0.2416
At 500 docs	0.1469
At 1000 docs	0.0941
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2857



Recall-Precision Curve

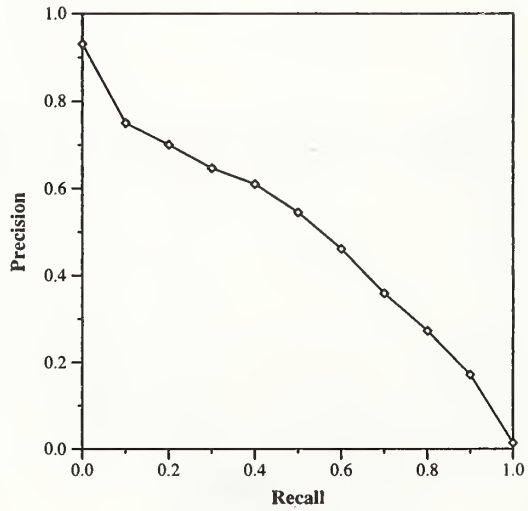


Difference from Median in Average Precision per Topic

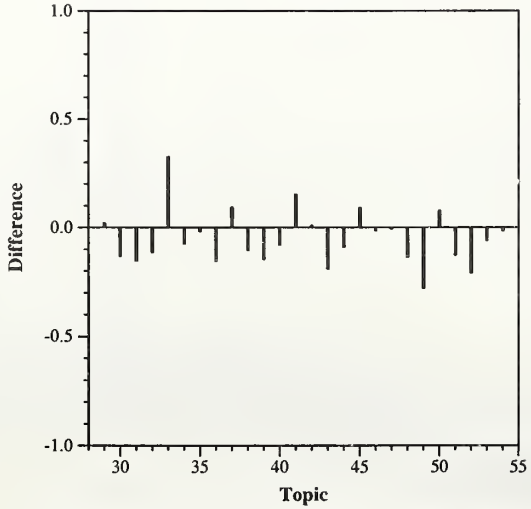
Summary Statistics	
Run Number	city97c1
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2495

Recall Level Precision Averages	
Recall	Precision
0.00	0.9312
0.10	0.7505
0.20	0.7008
0.30	0.6468
0.40	0.6103
0.50	0.5451
0.60	0.4614
0.70	0.3589
0.80	0.2727
0.90	0.1717
1.00	0.0138
Average precision over all relevant docs	
non-interpolated	0.4838

Document Level Averages	
	Precision
At 5 docs	0.7846
At 10 docs	0.7038
At 15 docs	0.6923
At 20 docs	0.6846
At 30 docs	0.6397
At 100 docs	0.4804
At 200 docs	0.3340
At 500 docs	0.1757
At 1000 docs	0.0960
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5119



Recall-Precision Curve

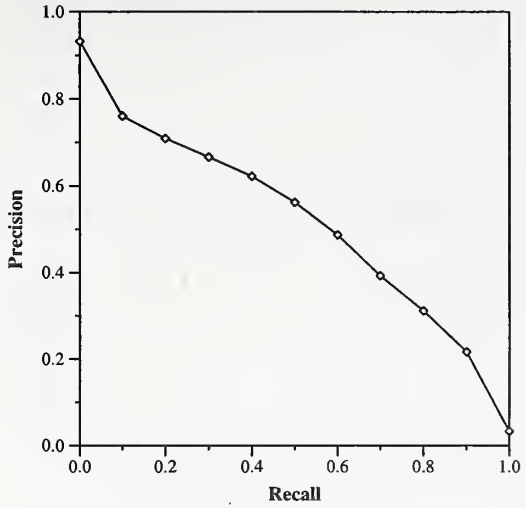


Difference from Median in Average Precision per Topic

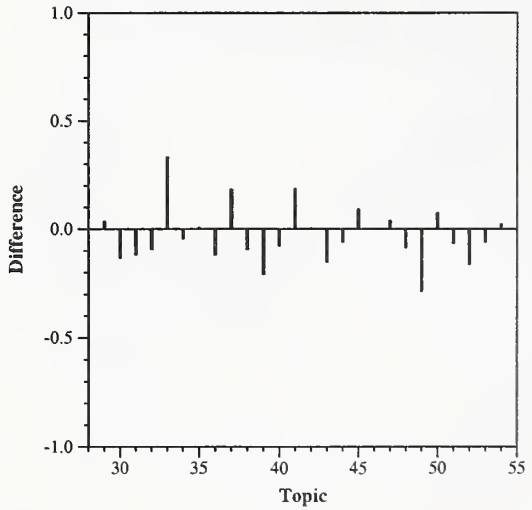
Summary Statistics	
Run Number	city97c2
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2589

Recall Level Precision Averages	
Recall	Precision
0.00	0.9325
0.10	0.7608
0.20	0.7094
0.30	0.6665
0.40	0.6220
0.50	0.5622
0.60	0.4872
0.70	0.3932
0.80	0.3118
0.90	0.2169
1.00	0.0330
Average precision over all relevant docs	
non-interpolated	0.5047

Document Level Averages	
	Precision
At 5 docs	0.7692
At 10 docs	0.7115
At 15 docs	0.7051
At 20 docs	0.6827
At 30 docs	0.6462
At 100 docs	0.4900
At 200 docs	0.3475
At 500 docs	0.1830
At 1000 docs	0.0996
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5221



Recall-Precision Curve

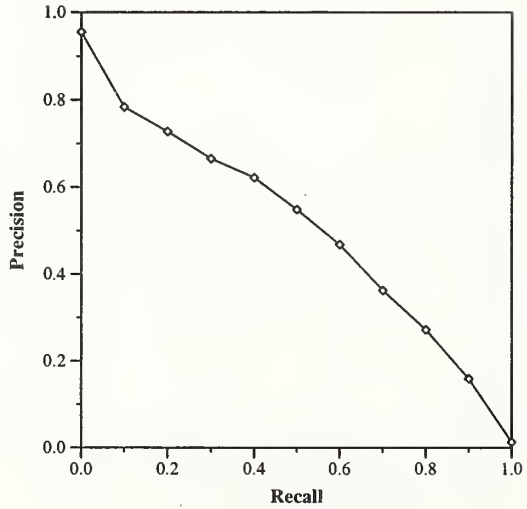


Difference from Median in Average Precision per Topic

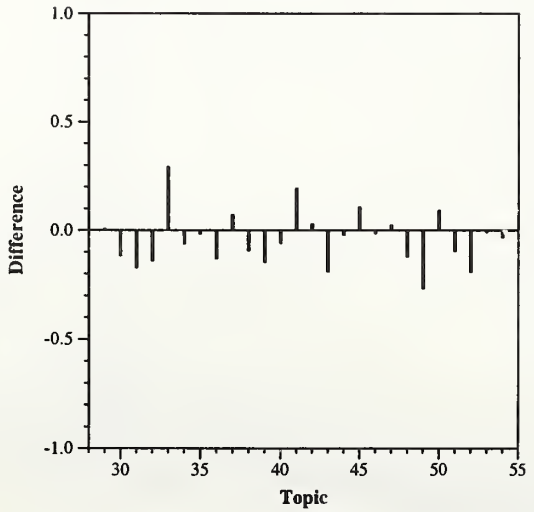
Summary Statistics	
Run Number	city97c3
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2461

Recall Level Precision Averages	
Recall	Precision
0.00	0.9552
0.10	0.7840
0.20	0.7279
0.30	0.6657
0.40	0.6221
0.50	0.5488
0.60	0.4684
0.70	0.3631
0.80	0.2724
0.90	0.1595
1.00	0.0133
Average precision over all relevant docs	
non-interpolated	0.4943

Document Level Averages	
	Precision
At 5 docs	0.8077
At 10 docs	0.7385
At 15 docs	0.7385
At 20 docs	0.7096
At 30 docs	0.6590
At 100 docs	0.4835
At 200 docs	0.3390
At 500 docs	0.1765
At 1000 docs	0.0947
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5178



Recall-Precision Curve

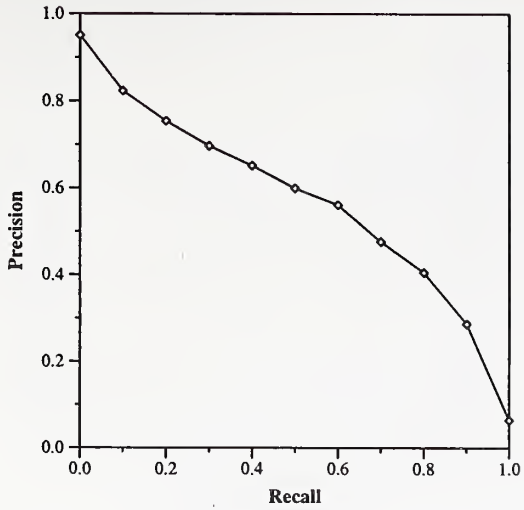


Difference from Median in Average Precision per Topic

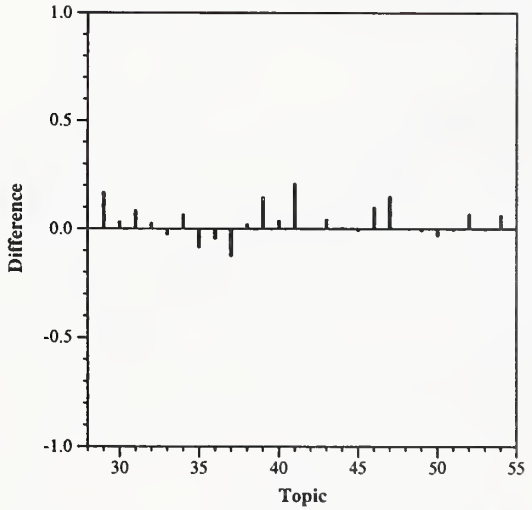
Summary Statistics	
Run Number	CLARITcAL
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2746

Recall Level Precision Averages	
Recall	Precision
0.00	0.9507
0.10	0.8236
0.20	0.7544
0.30	0.6971
0.40	0.6516
0.50	0.5995
0.60	0.5611
0.70	0.4765
0.80	0.4054
0.90	0.2868
1.00	0.0643
Average precision over all relevant docs	
non-interpolated	0.5683

Document Level Averages	
	Precision
At 5 docs	0.8538
At 10 docs	0.8115
At 15 docs	0.7872
At 20 docs	0.7635
At 30 docs	0.7128
At 100 docs	0.5115
At 200 docs	0.3717
At 500 docs	0.1954
At 1000 docs	0.1056
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5464



Recall-Precision Curve

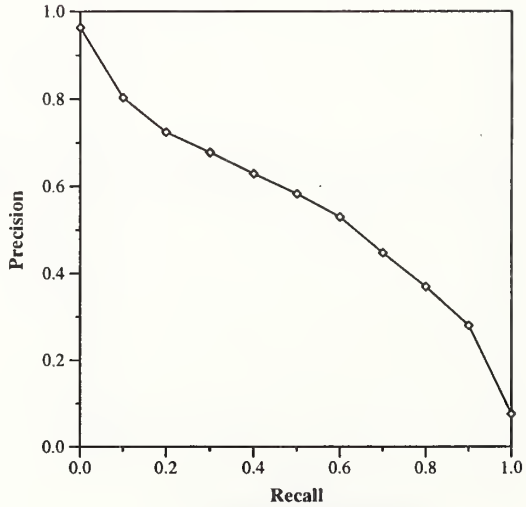


Difference from Median in Average Precision per Topic

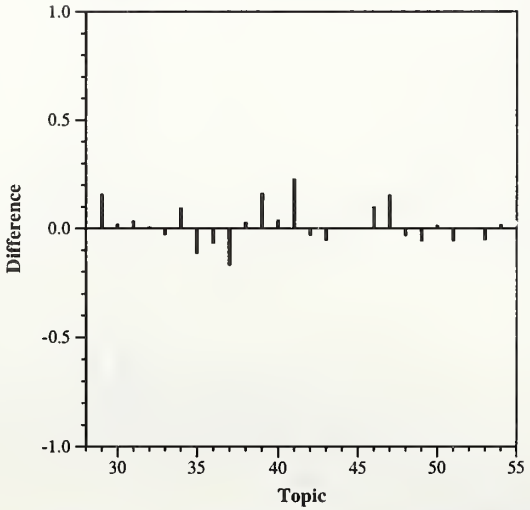
Summary Statistics	
Run Number	CLARITcAS
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2719

Recall Level Precision Averages	
Recall	Precision
0.00	0.9634
0.10	0.8034
0.20	0.7244
0.30	0.6779
0.40	0.6291
0.50	0.5828
0.60	0.5296
0.70	0.4470
0.80	0.3692
0.90	0.2797
1.00	0.0757
Average precision over all relevant docs	
non-interpolated	0.5494

Document Level Averages	
	Precision
At 5 docs	0.8615
At 10 docs	0.8154
At 15 docs	0.7897
At 20 docs	0.7442
At 30 docs	0.6885
At 100 docs	0.4938
At 200 docs	0.3652
At 500 docs	0.1902
At 1000 docs	0.1046
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5357



Recall-Precision Curve

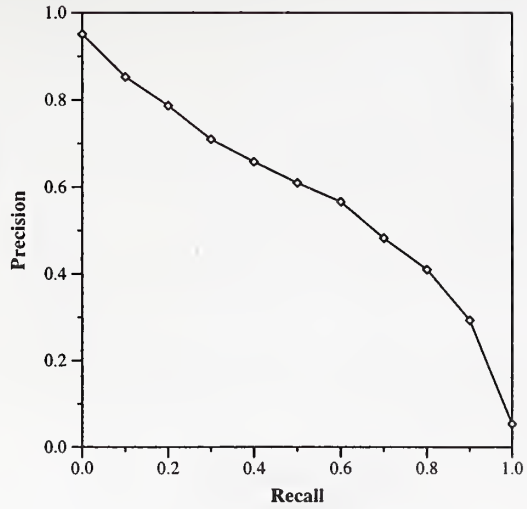


Difference from Median in Average Precision per Topic

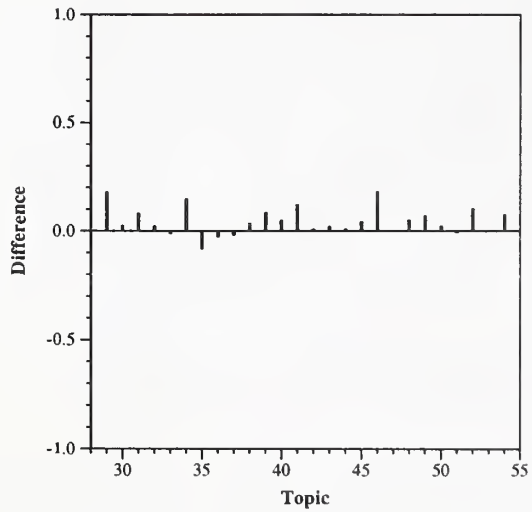
Summary Statistics	
Run Number	CLARITcM
Run Description	manual
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2774

Recall Level Precision Averages	
Recall	Precision
0.00	0.9512
0.10	0.8534
0.20	0.7875
0.30	0.7100
0.40	0.6585
0.50	0.6098
0.60	0.5662
0.70	0.4825
0.80	0.4103
0.90	0.2932
1.00	0.0539
Average precision over all relevant docs	
non-interpolated	0.5797

Document Level Averages	
	Precision
At 5 docs	0.8769
At 10 docs	0.8615
At 15 docs	0.8179
At 20 docs	0.7885
At 30 docs	0.7436
At 100 docs	0.5242
At 200 docs	0.3787
At 500 docs	0.1984
At 1000 docs	0.1067
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5475



Recall-Precision Curve

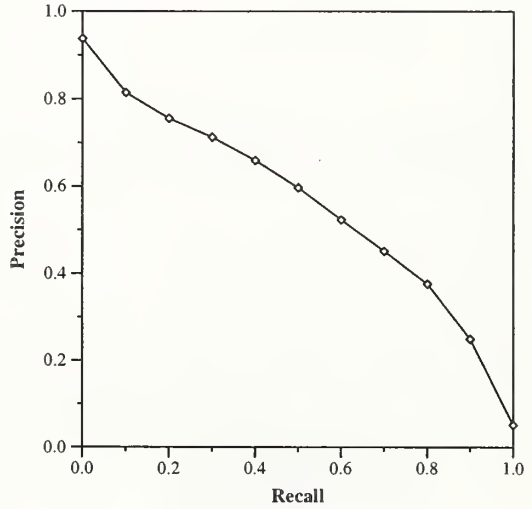


Difference from Median in Average Precision per Topic

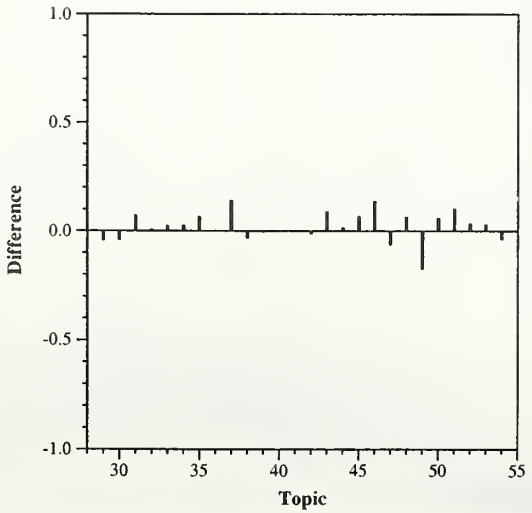
Summary Statistics	
Run Number	Cor6CH1sc
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2765

Recall Level Precision Averages	
Recall	Precision
0.00	0.9379
0.10	0.8146
0.20	0.7560
0.30	0.7129
0.40	0.6599
0.50	0.5972
0.60	0.5237
0.70	0.4518
0.80	0.3765
0.90	0.2498
1.00	0.0514
Average precision over all relevant docs	
non-interpolated	0.5547

Document Level Averages	
	Precision
At 5 docs	0.8154
At 10 docs	0.7923
At 15 docs	0.7744
At 20 docs	0.7423
At 30 docs	0.7115
At 100 docs	0.5162
At 200 docs	0.3754
At 500 docs	0.1976
At 1000 docs	0.1063
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5301



Recall-Precision Curve

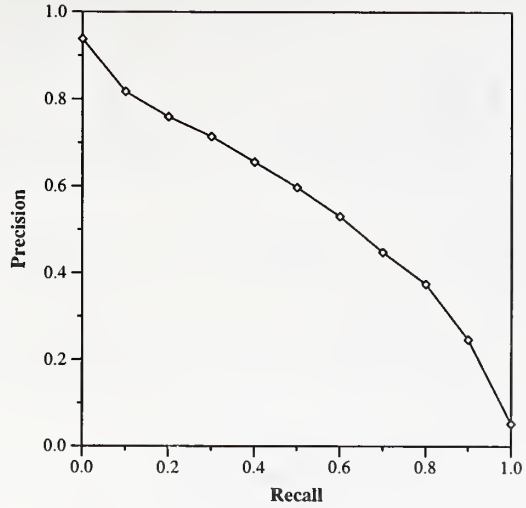


Difference from Median in Average Precision per Topic

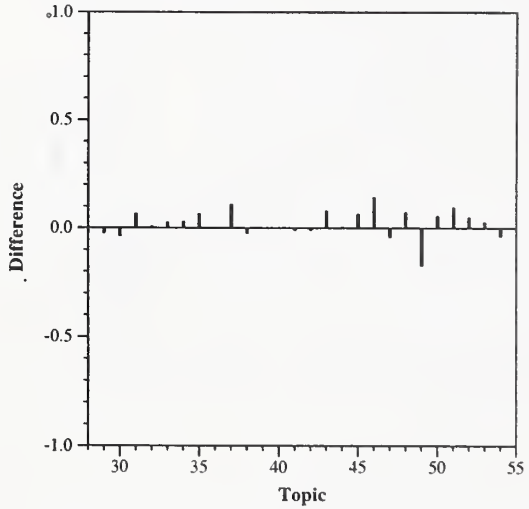
Summary Statistics	
Run Number	Cor6CH2ns
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2763

Recall Level Precision Averages	
Recall	Precision
0.00	0.9381
0.10	0.8174
0.20	0.7601
0.30	0.7146
0.40	0.6564
0.50	0.5977
0.60	0.5310
0.70	0.4486
0.80	0.3753
0.90	0.2471
1.00	0.0525
Average precision over all relevant docs	
non-interpolated	0.5552

Document Level Averages	
	Precision
At 5 docs	0.7923
At 10 docs	0.7923
At 15 docs	0.7872
At 20 docs	0.7442
At 30 docs	0.7115
At 100 docs	0.5185
At 200 docs	0.3765
At 500 docs	0.1968
At 1000 docs	0.1063
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5369



Recall-Precision Curve

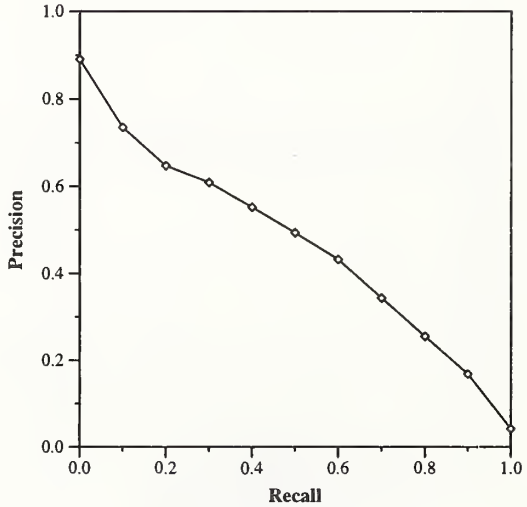


Difference from Median in Average Precision per Topic

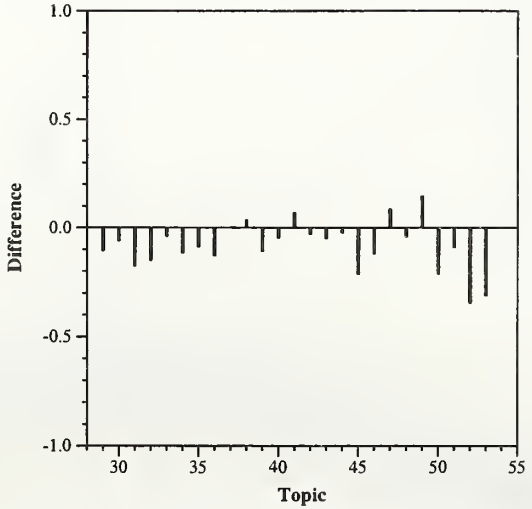
Summary Statistics	
Run Number	itich1
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2447

Recall Level Precision Averages	
Recall	Precision
0.00	0.8910
0.10	0.7352
0.20	0.6477
0.30	0.6092
0.40	0.5523
0.50	0.4936
0.60	0.4323
0.70	0.3437
0.80	0.2551
0.90	0.1688
1.00	0.0423
Average precision over all relevant docs	
non-interpolated	0.4541

Document Level Averages	
	Precision
At 5 docs	0.7385
At 10 docs	0.7192
At 15 docs	0.7026
At 20 docs	0.6577
At 30 docs	0.6115
At 100 docs	0.4615
At 200 docs	0.3158
At 500 docs	0.1644
At 1000 docs	0.0941
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4755



Recall-Precision Curve

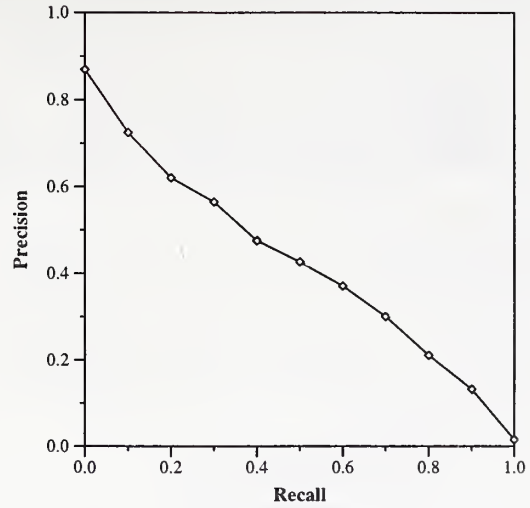


Difference from Median in Average Precision per Topic

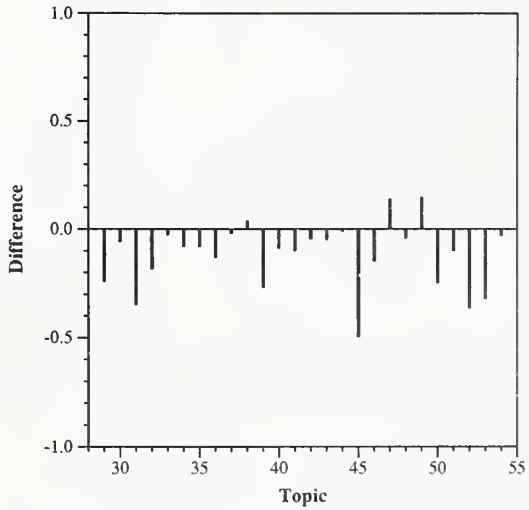
Summary Statistics	
Run Number	itich2
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2349

Recall Level Precision Averages	
Recall	Precision
0.00	0.8698
0.10	0.7248
0.20	0.6206
0.30	0.5649
0.40	0.4757
0.50	0.4264
0.60	0.3709
0.70	0.3005
0.80	0.2110
0.90	0.1321
1.00	0.0155
Average precision over all relevant docs	
non-interpolated	0.4145

Document Level Averages	
	Precision
At 5 docs	0.7538
At 10 docs	0.7192
At 15 docs	0.6872
At 20 docs	0.6500
At 30 docs	0.5885
At 100 docs	0.4288
At 200 docs	0.2973
At 500 docs	0.1590
At 1000 docs	0.0903
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4452



Recall-Precision Curve

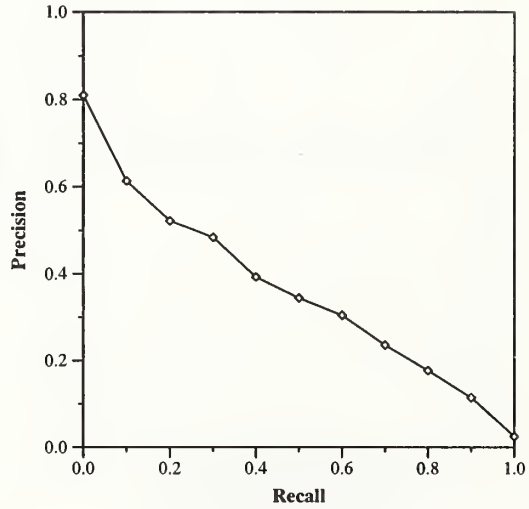


Difference from Median in Average Precision per Topic

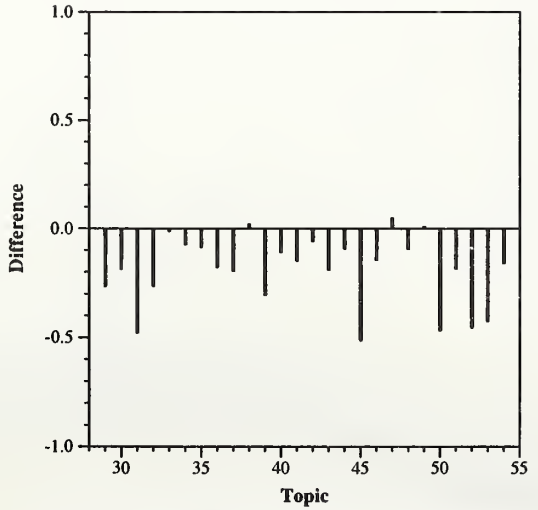
Summary Statistics	
Run Number	itich3
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2215

Recall Level Precision Averages	
Recall	Precision
0.00	0.8100
0.10	0.6137
0.20	0.5222
0.30	0.4842
0.40	0.3930
0.50	0.3446
0.60	0.3047
0.70	0.2362
0.80	0.1777
0.90	0.1148
1.00	0.0252
Average precision over all relevant docs	
non-interpolated	0.3427

Document Level Averages	
	Precision
At 5 docs	0.6385
At 10 docs	0.6038
At 15 docs	0.5821
At 20 docs	0.5692
At 30 docs	0.5051
At 100 docs	0.3715
At 200 docs	0.2671
At 500 docs	0.1501
At 1000 docs	0.0852
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3881



Recall-Precision Curve

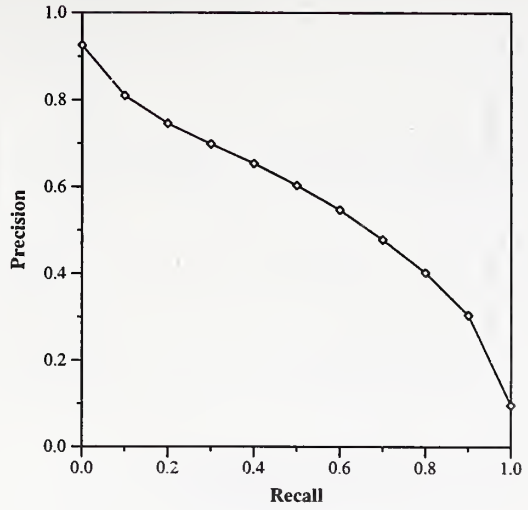


Difference from Median in Average Precision per Topic

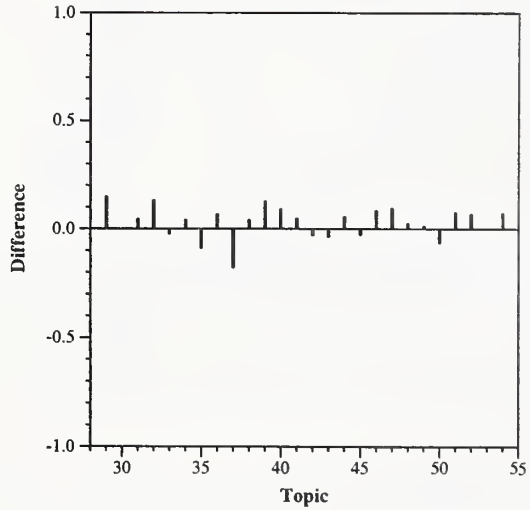
Summary Statistics	
Run Number	iss97CbD
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2802

Recall Level Precision Averages	
Recall	Precision
0.00	0.9255
0.10	0.8103
0.20	0.7463
0.30	0.6988
0.40	0.6541
0.50	0.6037
0.60	0.5471
0.70	0.4788
0.80	0.4027
0.90	0.3046
1.00	0.0967
Average precision over all relevant docs	
non-interpolated	0.5646

Document Level Averages	
	Precision
At 5 docs	0.8308
At 10 docs	0.8154
At 15 docs	0.7795
At 20 docs	0.7423
At 30 docs	0.7090
At 100 docs	0.5104
At 200 docs	0.3735
At 500 docs	0.1971
At 1000 docs	0.1078
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5515



Recall-Precision Curve

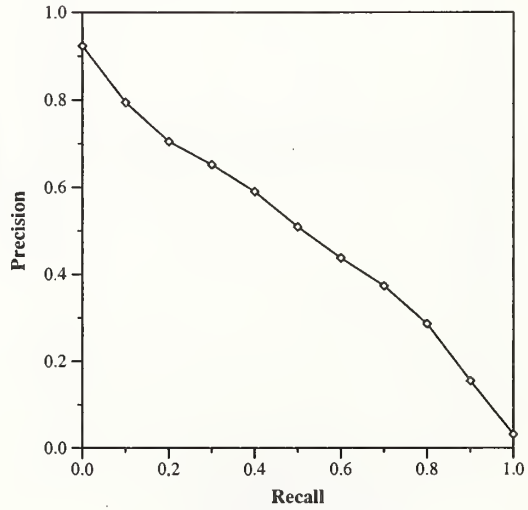


Difference from Median in Average Precision per Topic

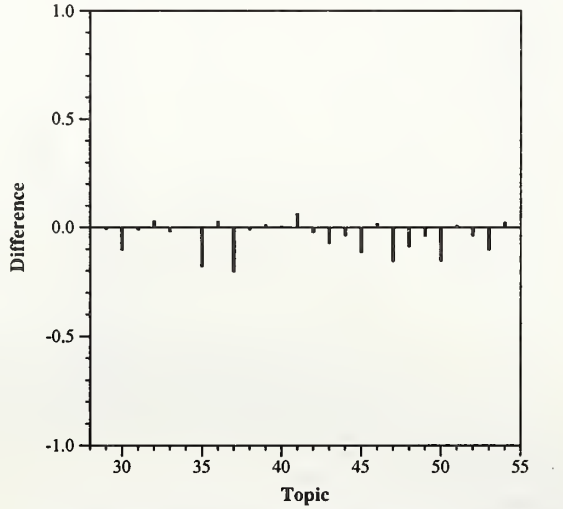
Summary Statistics	
Run Number	iss97CmD
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2723

Recall Level Precision Averages	
Recall	Precision
0.00	0.9237
0.10	0.7950
0.20	0.7055
0.30	0.6523
0.40	0.5906
0.50	0.5095
0.60	0.4380
0.70	0.3739
0.80	0.2864
0.90	0.1550
1.00	0.0316
Average precision over all relevant docs	
non-interpolated	0.4903

Document Level Averages	
	Precision
At 5 docs	0.8154
At 10 docs	0.7923
At 15 docs	0.7513
At 20 docs	0.7212
At 30 docs	0.6756
At 100 docs	0.4692
At 200 docs	0.3352
At 500 docs	0.1799
At 1000 docs	0.1047
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4941



Recall-Precision Curve

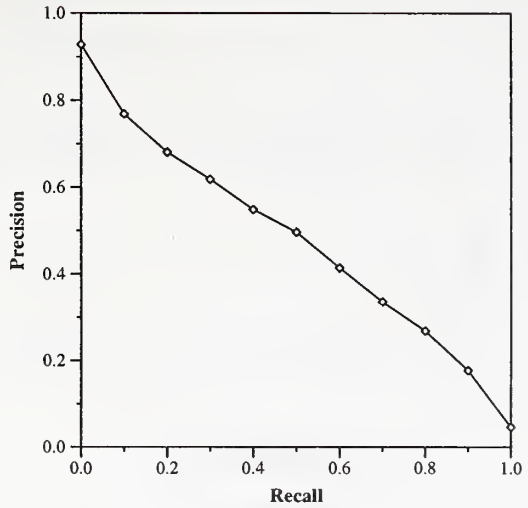


Difference from Median in Average Precision per Topic

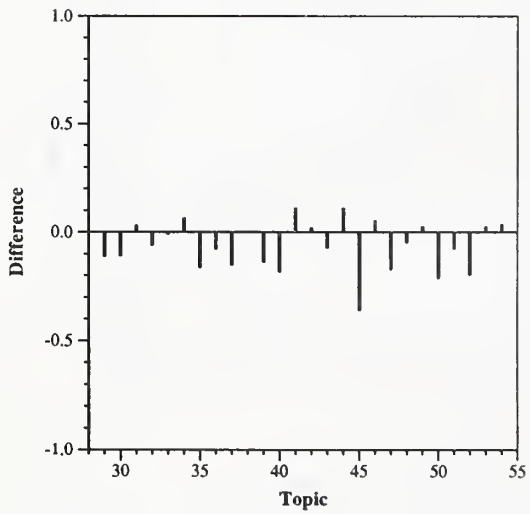
Summary Statistics	
Run Number	iss97CsD
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2619

Recall Level Precision Averages	
Recall	Precision
0.00	0.9279
0.10	0.7686
0.20	0.6808
0.30	0.6185
0.40	0.5489
0.50	0.4964
0.60	0.4142
0.70	0.3359
0.80	0.2689
0.90	0.1776
1.00	0.0463
Average precision over all relevant docs	
non-interpolated	0.4709

Document Level Averages	
	Precision
At 5 docs	0.7769
At 10 docs	0.7346
At 15 docs	0.7077
At 20 docs	0.6942
At 30 docs	0.6538
At 100 docs	0.4615
At 200 docs	0.3263
At 500 docs	0.1815
At 1000 docs	0.1007
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4689



Recall-Precision Curve

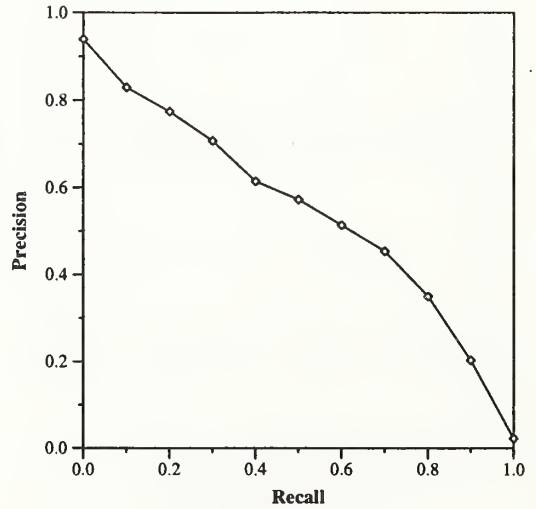


Difference from Median in Average Precision per Topic

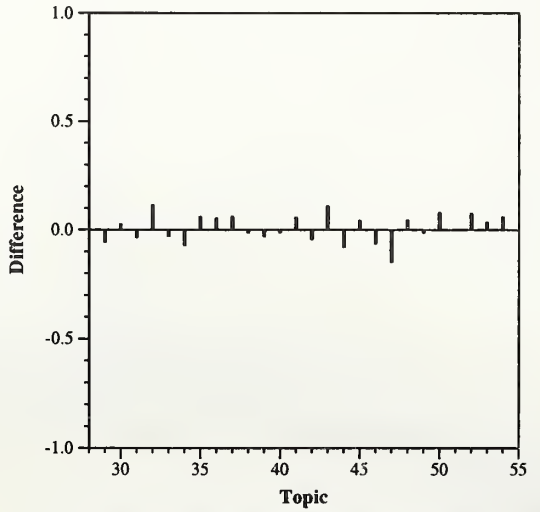
Summary Statistics	
Run Number	mds607
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2590

Recall Level Precision Averages	
Recall	Precision
0.00	0.9396
0.10	0.8297
0.20	0.7744
0.30	0.7075
0.40	0.6146
0.50	0.5727
0.60	0.5138
0.70	0.4541
0.80	0.3502
0.90	0.2033
1.00	0.0225
Average precision over all relevant docs	
non-interpolated	0.5436

Document Level Averages	
	Precision
At 5 docs	0.8769
At 10 docs	0.8192
At 15 docs	0.7897
At 20 docs	0.7577
At 30 docs	0.7051
At 100 docs	0.5112
At 200 docs	0.3687
At 500 docs	0.1911
At 1000 docs	0.0996
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5236



Recall-Precision Curve

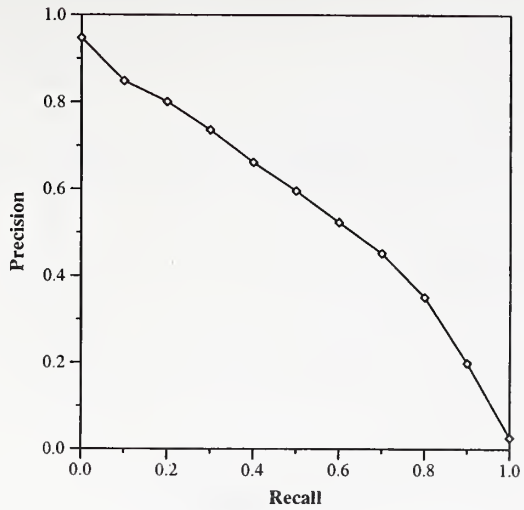


Difference from Median in Average Precision per Topic

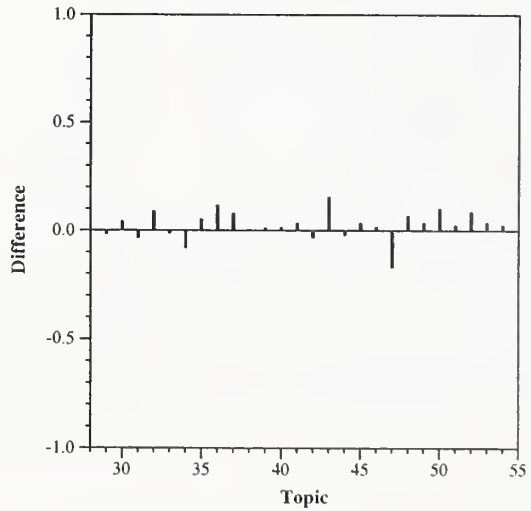
Summary Statistics	
Run Number	mds608
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2665

Recall Level Precision Averages	
Recall	Precision
0.00	0.9471
0.10	0.8489
0.20	0.8015
0.30	0.7366
0.40	0.6619
0.50	0.5959
0.60	0.5238
0.70	0.4525
0.80	0.3517
0.90	0.1994
1.00	0.0267
Average precision over all relevant docs	
non-interpolated	0.5597

Document Level Averages	
	Precision
At 5 docs	0.8692
At 10 docs	0.8231
At 15 docs	0.8051
At 20 docs	0.7731
At 30 docs	0.7385
At 100 docs	0.5165
At 200 docs	0.3760
At 500 docs	0.1932
At 1000 docs	0.1025
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5271



Recall-Precision Curve

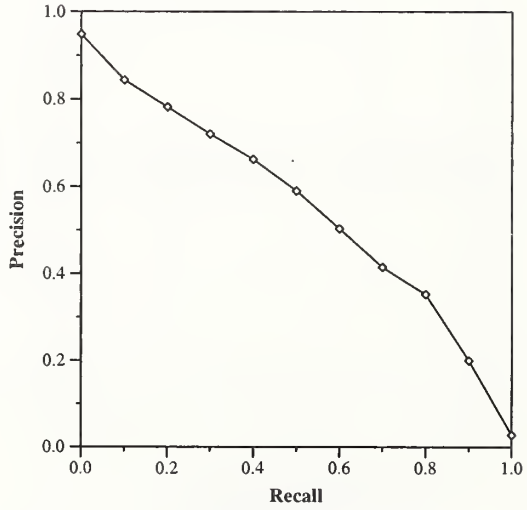


Difference from Median in Average Precision per Topic

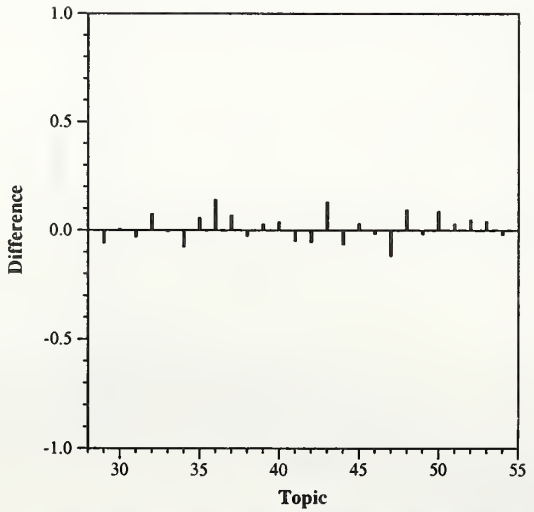
Summary Statistics	
Run Number	mds609
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2665

Recall Level Precision Averages	
Recall	Precision
0.00	0.9484
0.10	0.8442
0.20	0.7825
0.30	0.7207
0.40	0.6629
0.50	0.5903
0.60	0.5036
0.70	0.4150
0.80	0.3523
0.90	0.1997
1.00	0.0278
Average precision over all relevant docs	
non-interpolated	0.5479

Document Level Averages	
	Precision
At 5 docs	0.8538
At 10 docs	0.8269
At 15 docs	0.7949
At 20 docs	0.7654
At 30 docs	0.7321
At 100 docs	0.5131
At 200 docs	0.3662
At 500 docs	0.1927
At 1000 docs	0.1025
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5234



Recall-Precision Curve

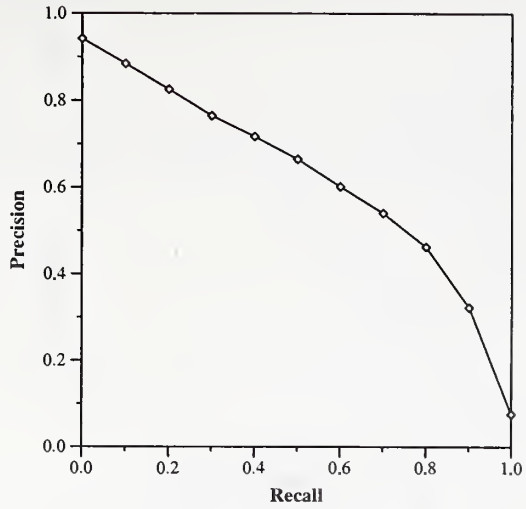


Difference from Median in Average Precision per Topic

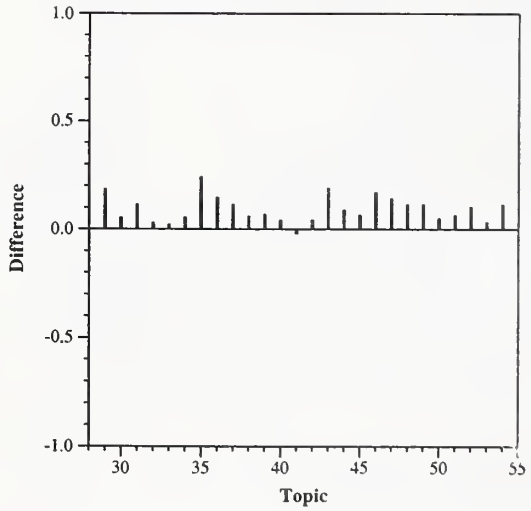
Summary Statistics	
Run Number	pirc7Ca
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	25999
Relevant:	2958
Rel-ret:	2795

Recall Level Precision Averages	
Recall	Precision
0.00	0.9422
0.10	0.8849
0.20	0.8258
0.30	0.7653
0.40	0.7174
0.50	0.6653
0.60	0.6019
0.70	0.5406
0.80	0.4626
0.90	0.3229
1.00	0.0759
Average precision over all relevant docs	
non-interpolated	0.6263

Document Level Averages	
	Precision
At 5 docs	0.8846
At 10 docs	0.8731
At 15 docs	0.8282
At 20 docs	0.8135
At 30 docs	0.7718
At 100 docs	0.5542
At 200 docs	0.4073
At 500 docs	0.2035
At 1000 docs	0.1075
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5809



Recall-Precision Curve



Difference from Median in Average Precision per Topic

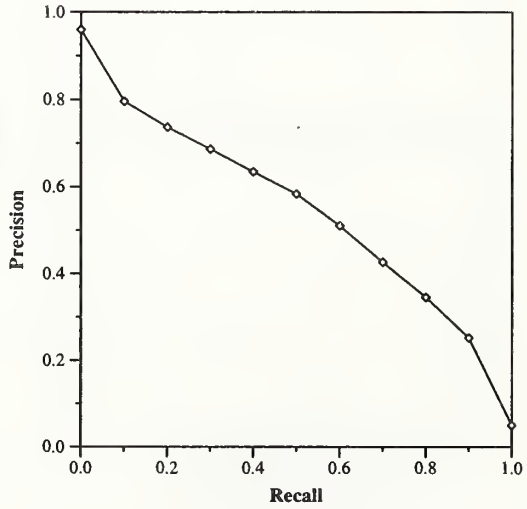
Summary Statistics	
Run Number	pirc7Cd
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	25999
Relevant:	2958
Rel-ret:	2674

Recall Level Precision Averages	
Recall	Precision
0.00	0.9598
0.10	0.7962
0.20	0.7371
0.30	0.6868
0.40	0.6345
0.50	0.5835
0.60	0.5104
0.70	0.4264
0.80	0.3459
0.90	0.2523
1.00	0.0501

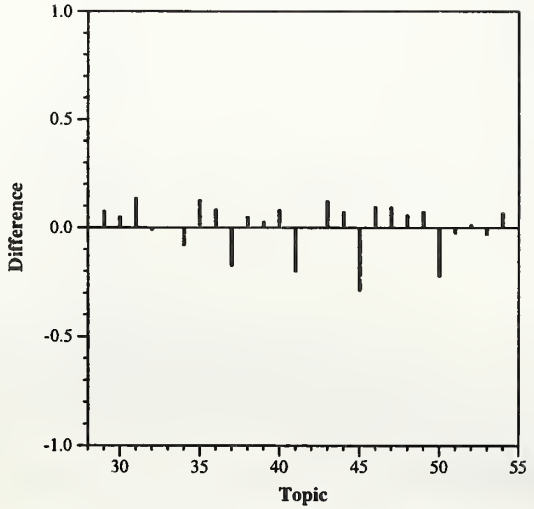
Average precision over all relevant docs	
non-interpolated	0.5423

Document Level Averages	
	Precision
At 5 docs	0.8615
At 10 docs	0.7962
At 15 docs	0.7846
At 20 docs	0.7519
At 30 docs	0.6974
At 100 docs	0.5035
At 200 docs	0.3642
At 500 docs	0.1893
At 1000 docs	0.1028

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5175



Recall-Precision Curve

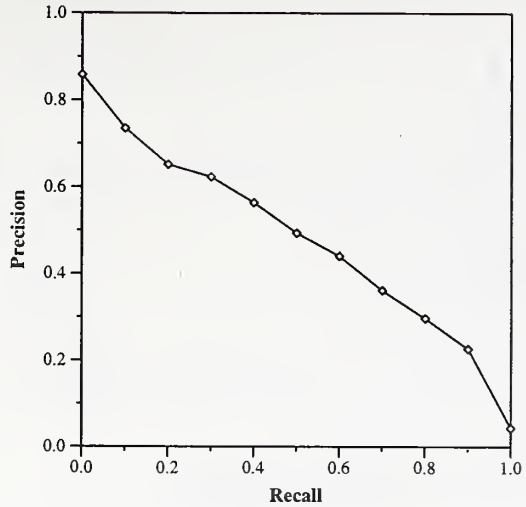


Difference from Median in Average Precision per Topic

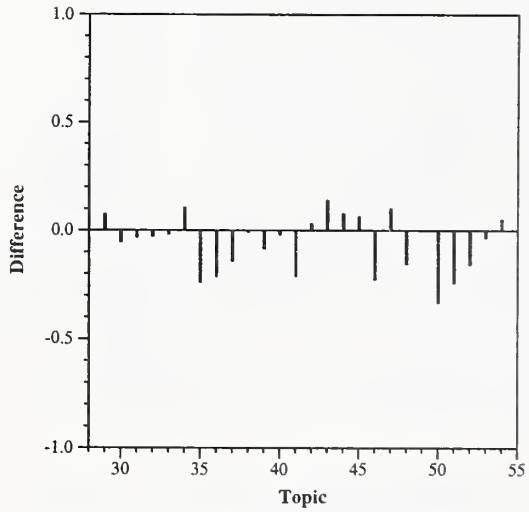
Summary Statistics	
Run Number	pirc7Ct
Run Description	automatic, title
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2547

Recall Level Precision Averages	
Recall	Precision
0.00	0.8586
0.10	0.7355
0.20	0.6520
0.30	0.6234
0.40	0.5644
0.50	0.4942
0.60	0.4408
0.70	0.3617
0.80	0.2973
0.90	0.2268
1.00	0.0437
Average precision over all relevant docs	
non-interpolated	0.4755

Document Level Averages	
	Precision
At 5 docs	0.7231
At 10 docs	0.7115
At 15 docs	0.6949
At 20 docs	0.6692
At 30 docs	0.6192
At 100 docs	0.4327
At 200 docs	0.3215
At 500 docs	0.1747
At 1000 docs	0.0980
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4630



Recall-Precision Curve

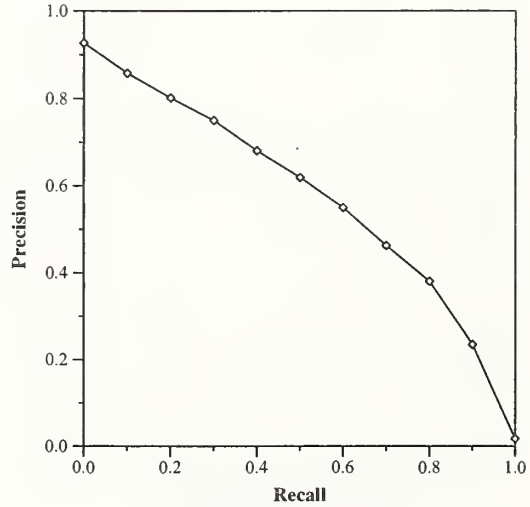


Difference from Median in Average Precision per Topic

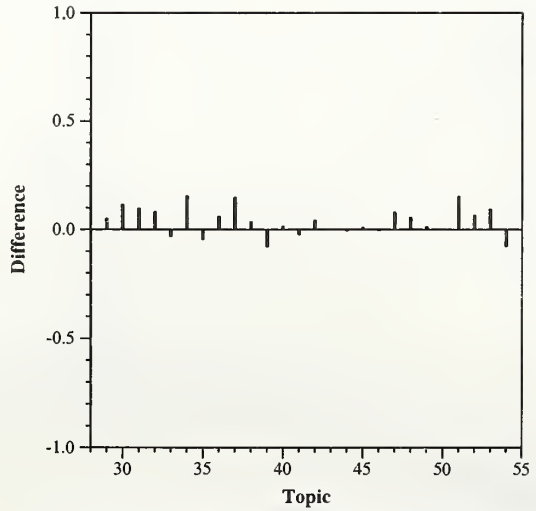
Summary Statistics	
Run Number	ETHccA
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2698

Recall Level Precision Averages	
Recall	Precision
0.00	0.9272
0.10	0.8585
0.20	0.8018
0.30	0.7501
0.40	0.6808
0.50	0.6193
0.60	0.5500
0.70	0.4628
0.80	0.3805
0.90	0.2349
1.00	0.0174
Average precision over all relevant docs	
non-interpolated	0.5733

Document Level Averages	
	Precision
At 5 docs	0.8615
At 10 docs	0.8269
At 15 docs	0.8103
At 20 docs	0.7788
At 30 docs	0.7513
At 100 docs	0.5281
At 200 docs	0.3817
At 500 docs	0.1938
At 1000 docs	0.1038
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5598



Recall-Precision Curve



Difference from Median in Average Precision per Topic

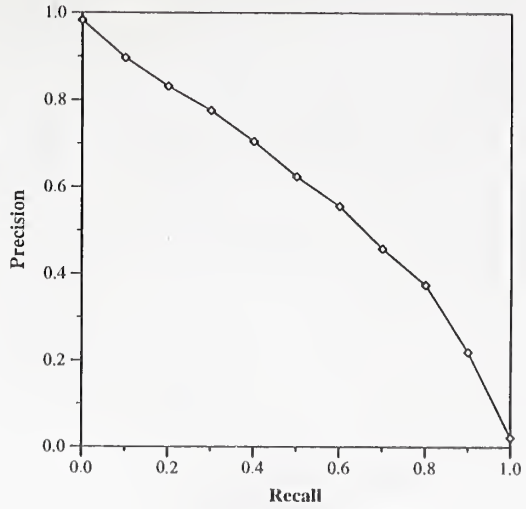
Summary Statistics	
Run Number	ETHccM
Run Description	manual
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2689

Recall Level Precision Averages	
Recall	Precision
0.00	0.9826
0.10	0.8970
0.20	0.8316
0.30	0.7761
0.40	0.7052
0.50	0.6242
0.60	0.5560
0.70	0.4586
0.80	0.3750
0.90	0.2203
1.00	0.0223

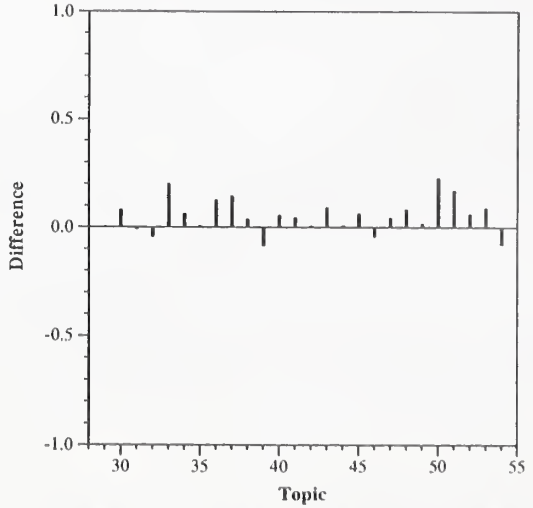
Average precision over all relevant docs	
non-interpolated	0.5868

Document Level Averages	
	Precision
At 5 docs	0.9231
At 10 docs	0.8769
At 15 docs	0.8308
At 20 docs	0.8192
At 30 docs	0.7705
At 100 docs	0.5342
At 200 docs	0.3765
At 500 docs	0.1926
At 1000 docs	0.1034

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5617



Recall-Precision Curve

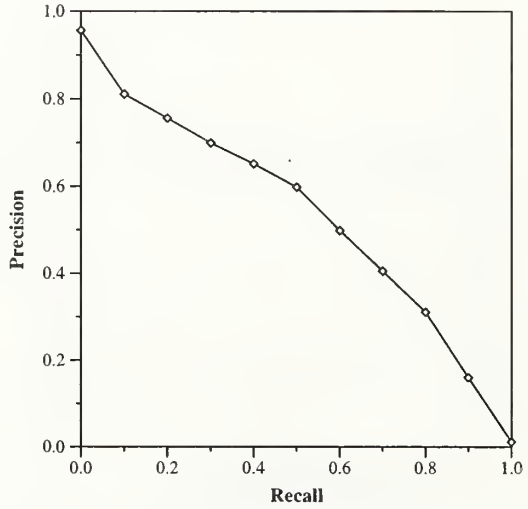


Difference from Median in Average Precision per Topic

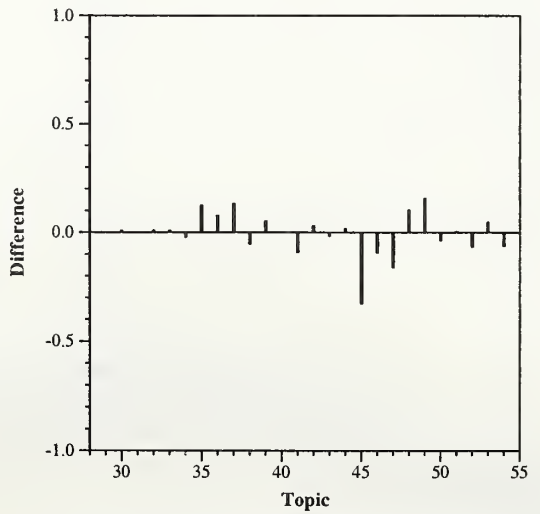
Summary Statistics	
Run Number	BrklyCH3
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2551

Recall Level Precision Averages	
Recall	Precision
0.00	0.9565
0.10	0.8112
0.20	0.7563
0.30	0.6994
0.40	0.6517
0.50	0.5980
0.60	0.4986
0.70	0.4055
0.80	0.3109
0.90	0.1607
1.00	0.0116
Average precision over all relevant docs	
non-interpolated	0.5291

Document Level Averages	
	Precision
At 5 docs	0.8308
At 10 docs	0.8038
At 15 docs	0.7667
At 20 docs	0.7519
At 30 docs	0.7103
At 100 docs	0.5269
At 200 docs	0.3635
At 500 docs	0.1820
At 1000 docs	0.0981
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5252



Recall-Precision Curve

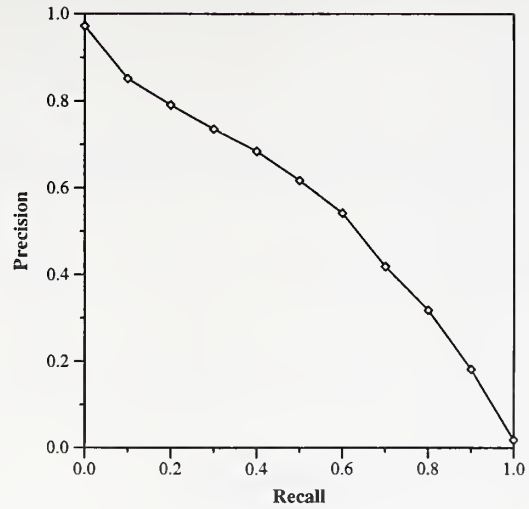


Difference from Median in Average Precision per Topic

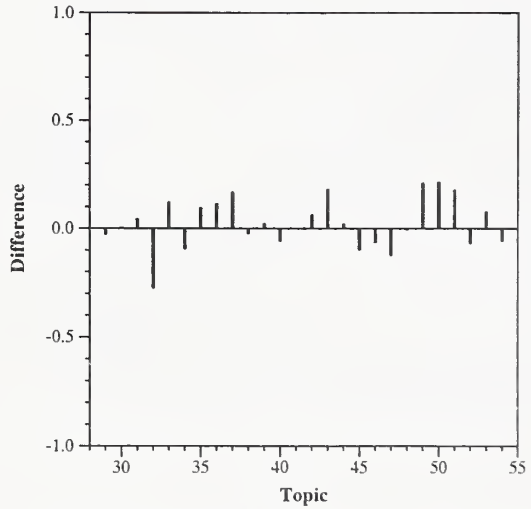
Summary Statistics	
Run Number	BrklyCH4
Run Description	manual
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2573

Recall Level Precision Averages	
Recall	Precision
0.00	0.9720
0.10	0.8516
0.20	0.7913
0.30	0.7359
0.40	0.6847
0.50	0.6177
0.60	0.5423
0.70	0.4194
0.80	0.3182
0.90	0.1818
1.00	0.0182
Average precision over all relevant docs	
non-interpolated	0.5586

Document Level Averages	
	Precision
At 5 docs	0.9231
At 10 docs	0.8346
At 15 docs	0.8205
At 20 docs	0.7904
At 30 docs	0.7372
At 100 docs	0.5427
At 200 docs	0.3637
At 500 docs	0.1820
At 1000 docs	0.0990
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5496



Recall-Precision Curve

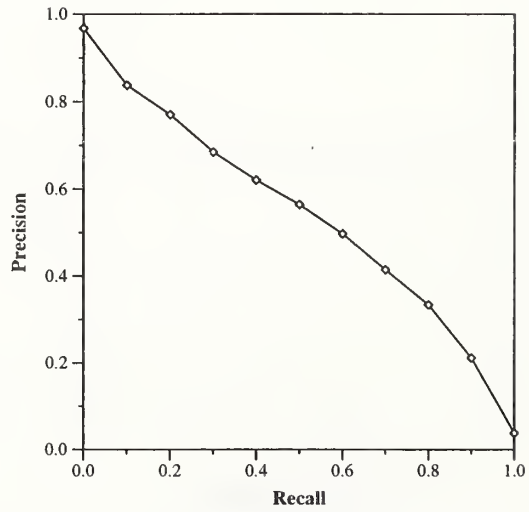


Difference from Median in Average Precision per Topic

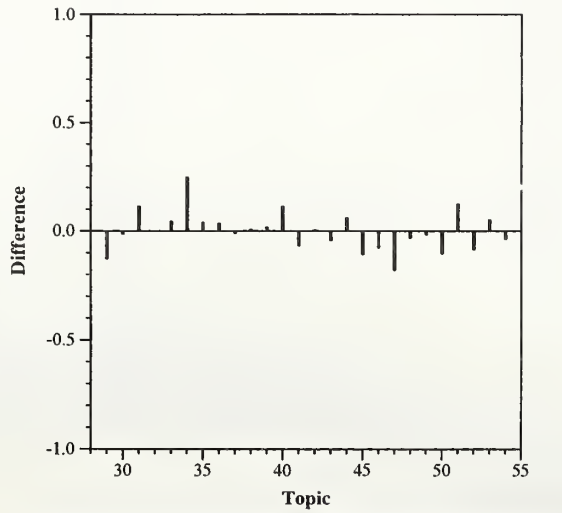
Summary Statistics	
Run Number	INQ4ch1
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2662

Recall Level Precision Averages	
Recall	Precision
0.00	0.9678
0.10	0.8381
0.20	0.7711
0.30	0.6851
0.40	0.6207
0.50	0.5642
0.60	0.4971
0.70	0.4146
0.80	0.3341
0.90	0.2119
1.00	0.0386
Average precision over all relevant docs	
non-interpolated	0.5336

Document Level Averages	
	Precision
At 5 docs	0.8308
At 10 docs	0.8077
At 15 docs	0.8000
At 20 docs	0.7654
At 30 docs	0.7141
At 100 docs	0.5096
At 200 docs	0.3615
At 500 docs	0.1879
At 1000 docs	0.1024
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5218



Recall-Precision Curve

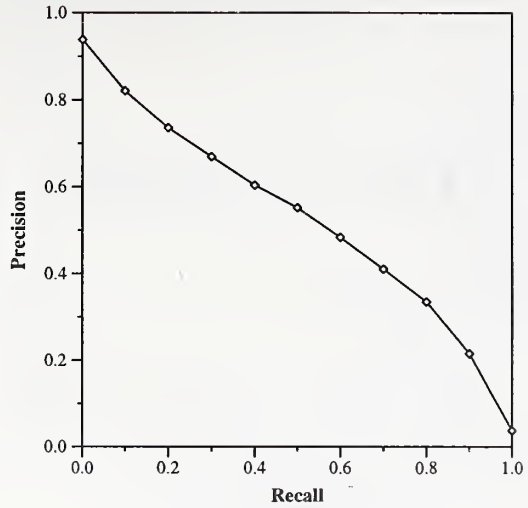


Difference from Median in Average Precision per Topic

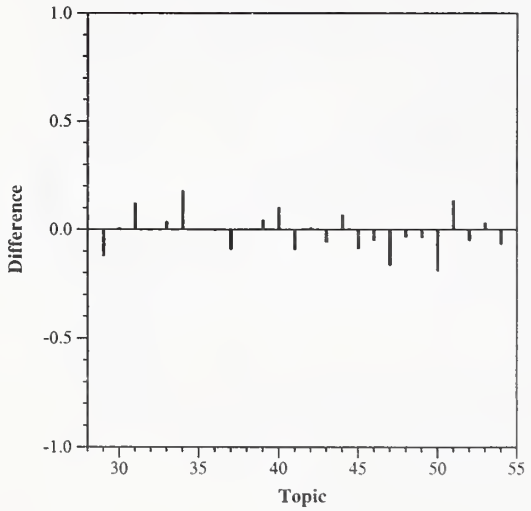
Summary Statistics	
Run Number	INQ4ch2
Run Description	automatic, short
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2664

Recall Level Precision Averages	
Recall	Precision
0.00	0.9386
0.10	0.8211
0.20	0.7366
0.30	0.6701
0.40	0.6043
0.50	0.5522
0.60	0.4840
0.70	0.4108
0.80	0.3352
0.90	0.2157
1.00	0.0376
Average precision over all relevant docs	
non-interpolated	0.5223

Document Level Averages	
	Precision
At 5 docs	0.8385
At 10 docs	0.8154
At 15 docs	0.7718
At 20 docs	0.7538
At 30 docs	0.7051
At 100 docs	0.4996
At 200 docs	0.3592
At 500 docs	0.1885
At 1000 docs	0.1025
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.5137



Recall-Precision Curve

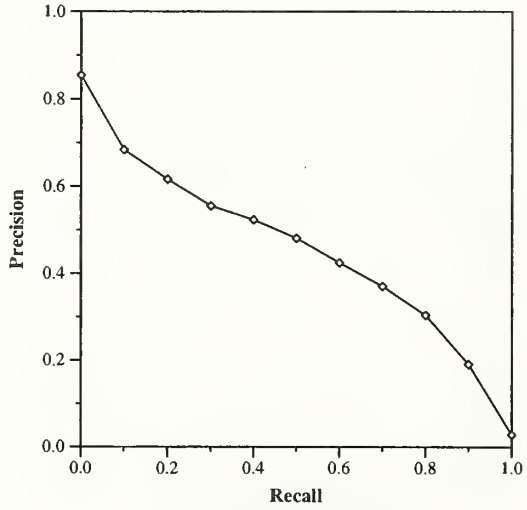


Difference from Median in Average Precision per Topic

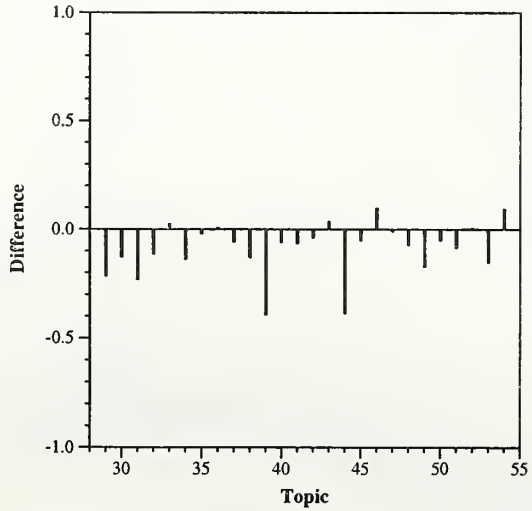
Summary Statistics	
Run Number	UdeMbi
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2709

Recall Level Precision Averages	
Recall	Precision
0.00	0.8546
0.10	0.6844
0.20	0.6165
0.30	0.5556
0.40	0.5236
0.50	0.4815
0.60	0.4252
0.70	0.3711
0.80	0.3043
0.90	0.1912
1.00	0.0284
Average precision over all relevant docs	
non-interpolated	0.4467

Document Level Averages	
	Precision
At 5 docs	0.6615
At 10 docs	0.7000
At 15 docs	0.6359
At 20 docs	0.5981
At 30 docs	0.5705
At 100 docs	0.4408
At 200 docs	0.3379
At 500 docs	0.1899
At 1000 docs	0.1042
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4655



Recall-Precision Curve

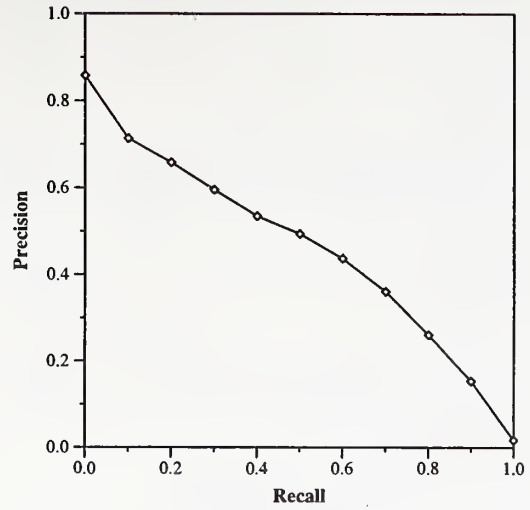


Difference from Median in Average Precision per Topic

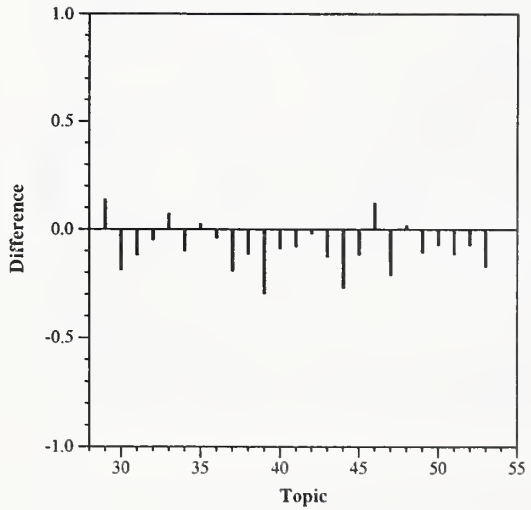
Summary Statistics	
Run Number	UdeMseg
Run Description	automatic, long
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2668

Recall Level Precision Averages	
Recall	Precision
0.00	0.8581
0.10	0.7134
0.20	0.6582
0.30	0.5953
0.40	0.5345
0.50	0.4939
0.60	0.4371
0.70	0.3609
0.80	0.2608
0.90	0.1541
1.00	0.0169
Average precision over all relevant docs	
non-interpolated	0.4524

Document Level Averages	
	Precision
At 5 docs	0.6769
At 10 docs	0.6423
At 15 docs	0.6333
At 20 docs	0.6173
At 30 docs	0.5782
At 100 docs	0.4662
At 200 docs	0.3402
At 500 docs	0.1857
At 1000 docs	0.1026
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4748



Recall-Precision Curve

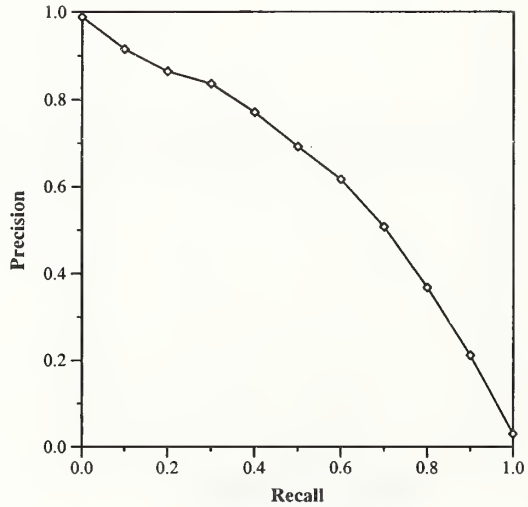


Difference from Median in Average Precision per Topic

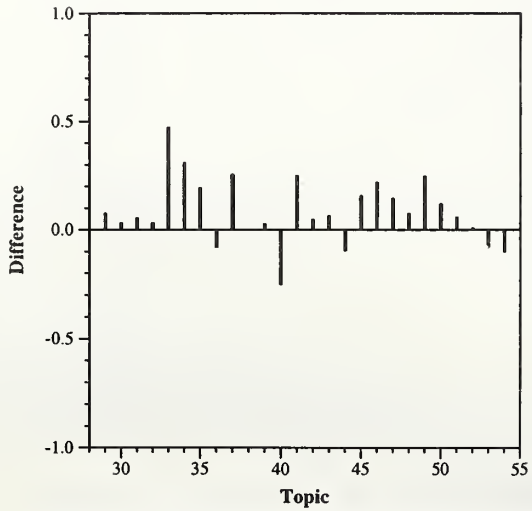
Summary Statistics	
Run Number	uwmt6c0
Run Description	manual
Number of Topics	26
Total number of documents over all topics	
Retrieved:	26000
Relevant:	2958
Rel-ret:	2552

Recall Level Precision Averages	
Recall	Precision
0.00	0.9886
0.10	0.9152
0.20	0.8645
0.30	0.8359
0.40	0.7706
0.50	0.6919
0.60	0.6168
0.70	0.5077
0.80	0.3680
0.90	0.2120
1.00	0.0300
Average precision over all relevant docs	
non-interpolated	0.6203

Document Level Averages	
	Precision
At 5 docs	0.9231
At 10 docs	0.8923
At 15 docs	0.8692
At 20 docs	0.8423
At 30 docs	0.7962
At 100 docs	0.5988
At 200 docs	0.3737
At 500 docs	0.1795
At 1000 docs	0.0982
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.6224



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	CEAef
Run Description	Cross-language run: English topics, French documents [unjudged]
Number of Topics	21

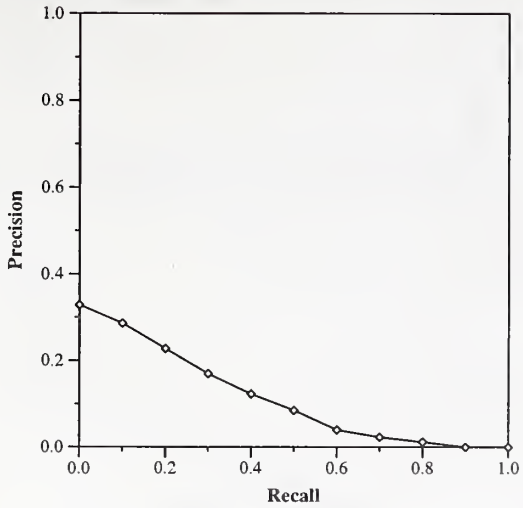
Total number of documents over all topics	
Retrieved:	19918
Relevant:	1239
Rel-ret:	577

Recall Level Precision Averages	
Recall	Precision
0.00	0.3289
0.10	0.2866
0.20	0.2282
0.30	0.1700
0.40	0.1231
0.50	0.0852
0.60	0.0400
0.70	0.0230
0.80	0.0117
0.90	0.0000
1.00	0.0000

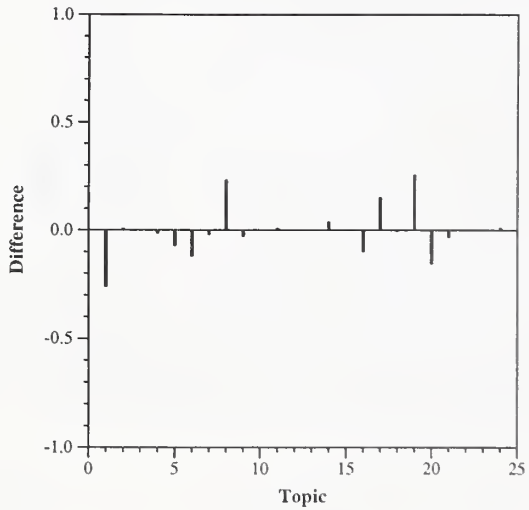
Average precision over all relevant docs	
non-interpolated	0.1065

Document Level Averages	
	Precision
At 5 docs	0.2286
At 10 docs	0.2190
At 15 docs	0.2190
At 20 docs	0.2095
At 30 docs	0.1873
At 100 docs	0.1271
At 200 docs	0.0895
At 500 docs	0.0492
At 1000 docs	0.0275

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1387



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	CEAff
Run Description	Monolingual French run [judged]
Number of Topics	21

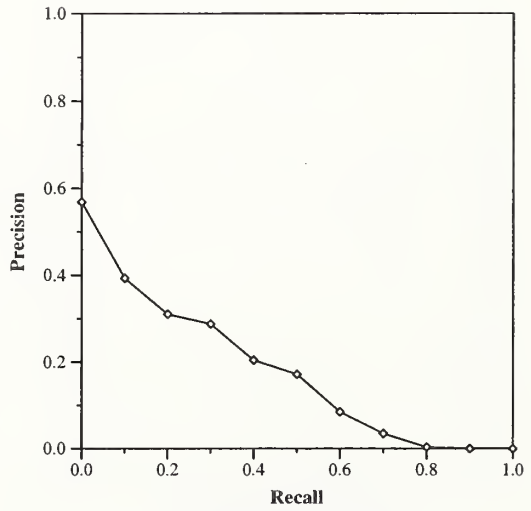
Total number of documents over all topics	
Retrieved:	20672
Relevant:	1239
Rel-ret:	612

Recall Level Precision Averages	
Recall	Precision
0.00	0.5687
0.10	0.3936
0.20	0.3106
0.30	0.2882
0.40	0.2046
0.50	0.1719
0.60	0.0850
0.70	0.0349
0.80	0.0033
0.90	0.0000
1.00	0.0000

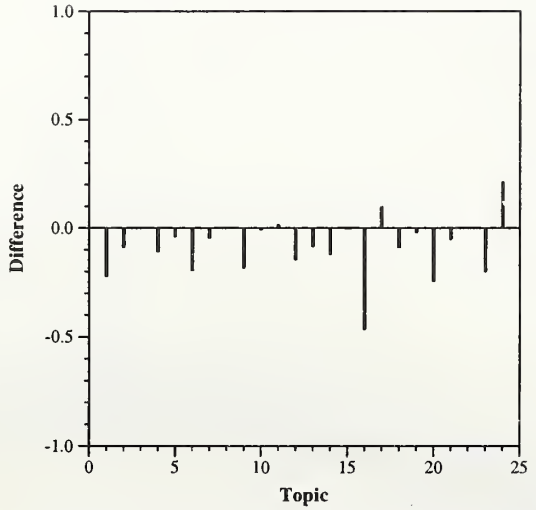
Average precision over all relevant docs	
non-interpolated	0.1626

Document Level Averages	
	Precision
At 5 docs	0.3810
At 10 docs	0.3524
At 15 docs	0.3365
At 20 docs	0.3000
At 30 docs	0.2635
At 100 docs	0.1519
At 200 docs	0.0974
At 500 docs	0.0492
At 1000 docs	0.0291

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2144



Recall-Precision Curve

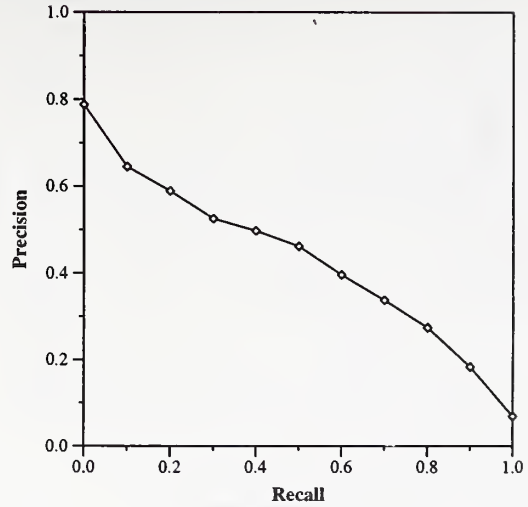


Difference from Median in Average Precision per Topic

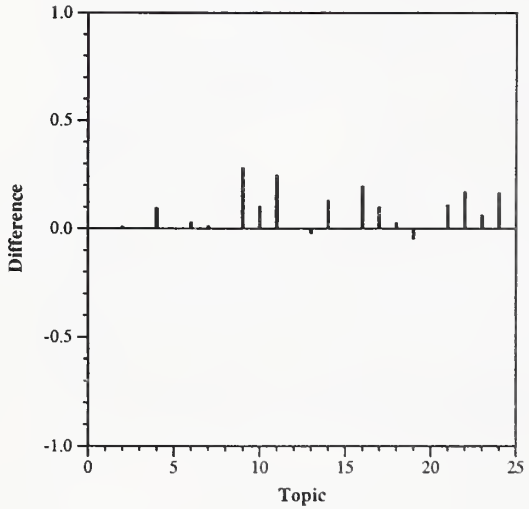
Summary Statistics	
Run Number	Cor6EEsc
Run Description	Monolingual English run [judged]
Number of Topics	21
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	1159

Recall Level Precision Averages	
Recall	Precision
0.00	0.7881
0.10	0.6454
0.20	0.5894
0.30	0.5258
0.40	0.4977
0.50	0.4617
0.60	0.3961
0.70	0.3376
0.80	0.2742
0.90	0.1838
1.00	0.0690
Average precision over all relevant docs	
non-interpolated	0.4202

Document Level Averages	
	Precision
At 5 docs	0.5714
At 10 docs	0.5429
At 15 docs	0.5397
At 20 docs	0.5167
At 30 docs	0.4698
At 100 docs	0.3376
At 200 docs	0.2160
At 500 docs	0.1033
At 1000 docs	0.0552
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4258



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Cor6EFexp
Run Description	Cross-language run: English topics, French documents [unjudged]
Number of Topics	21

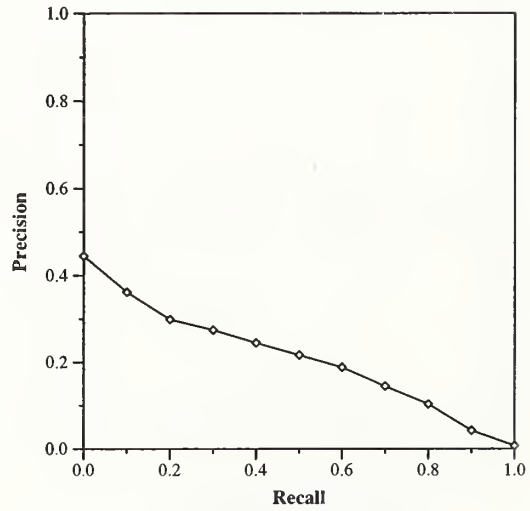
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	689

Recall Level Precision Averages	
Recall	Precision
0.00	0.4451
0.10	0.3622
0.20	0.2990
0.30	0.2748
0.40	0.2449
0.50	0.2167
0.60	0.1884
0.70	0.1448
0.80	0.1037
0.90	0.0425
1.00	0.0072

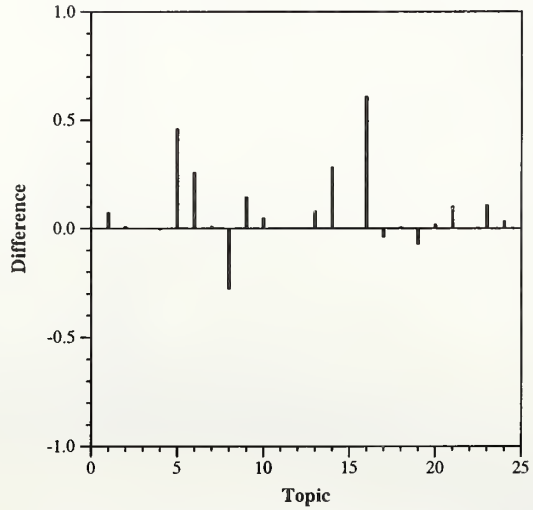
Average precision over all relevant docs	
non-interpolated	0.1982

Document Level Averages	
	Precision
At 5 docs	0.3143
At 10 docs	0.3381
At 15 docs	0.2857
At 20 docs	0.2619
At 30 docs	0.2429
At 100 docs	0.1548
At 200 docs	0.1067
At 500 docs	0.0567
At 1000 docs	0.0328

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2343



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Cor6FFsc
Run Description	Monolingual French run [judged]
Number of Topics	21

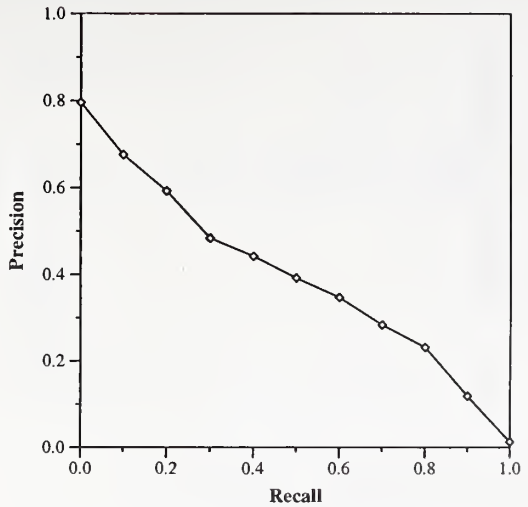
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	1078

Recall Level Precision Averages	
Recall	Precision
0.00	0.7963
0.10	0.6763
0.20	0.5927
0.30	0.4838
0.40	0.4422
0.50	0.3921
0.60	0.3470
0.70	0.2838
0.80	0.2318
0.90	0.1188
1.00	0.0128

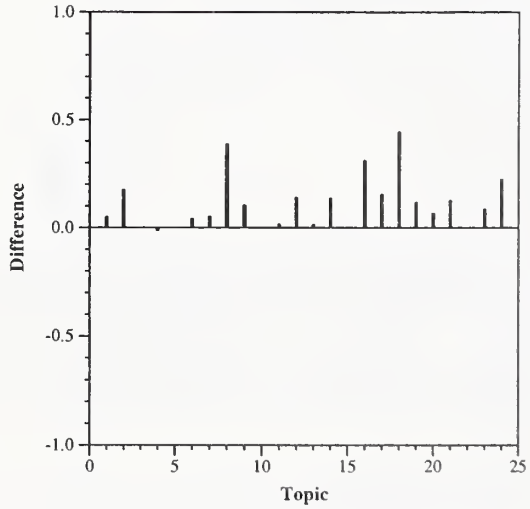
Average precision over all relevant docs	
non-interpolated	0.3815

Document Level Averages	
	Precision
At 5 docs	0.6286
At 10 docs	0.5190
At 15 docs	0.5016
At 20 docs	0.4667
At 30 docs	0.4159
At 100 docs	0.2695
At 200 docs	0.1857
At 500 docs	0.0943
At 1000 docs	0.0513

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4042



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	DCU97Fv2
Run Description	Monolingual French run [judged]
Number of Topics	21

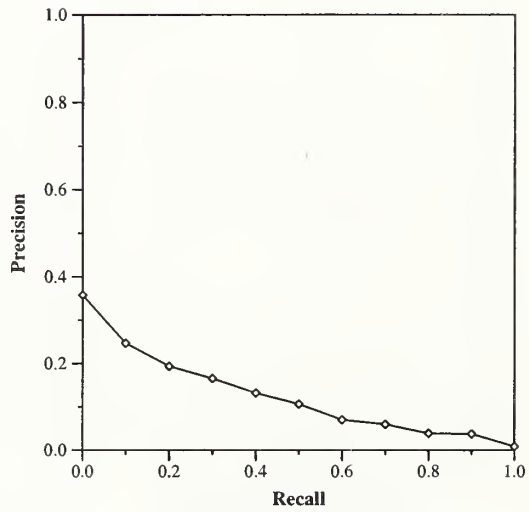
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	549

Recall Level Precision Averages	
Recall	Precision
0.00	0.3580
0.10	0.2471
0.20	0.1940
0.30	0.1657
0.40	0.1322
0.50	0.1067
0.60	0.0702
0.70	0.0600
0.80	0.0388
0.90	0.0377
1.00	0.0087

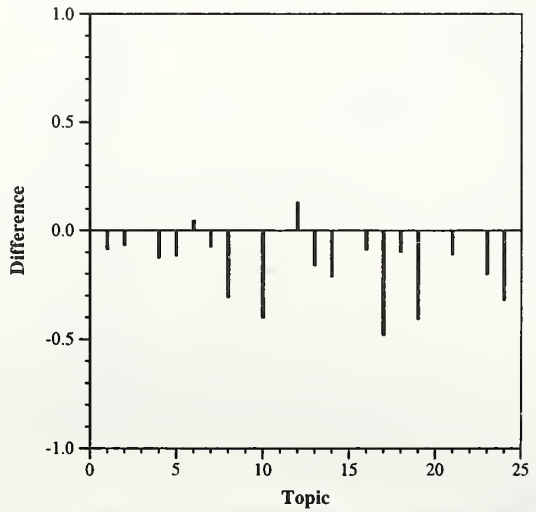
Average precision over all relevant docs	
non-interpolated	0.1102

Document Level Averages	
	Precision
At 5 docs	0.1905
At 10 docs	0.2095
At 15 docs	0.1905
At 20 docs	0.1857
At 30 docs	0.1730
At 100 docs	0.1190
At 200 docs	0.0769
At 500 docs	0.0433
At 1000 docs	0.0261

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1418



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	97lsiLEE
Run Description	Monolingual English run [judged]
Number of Topics	21

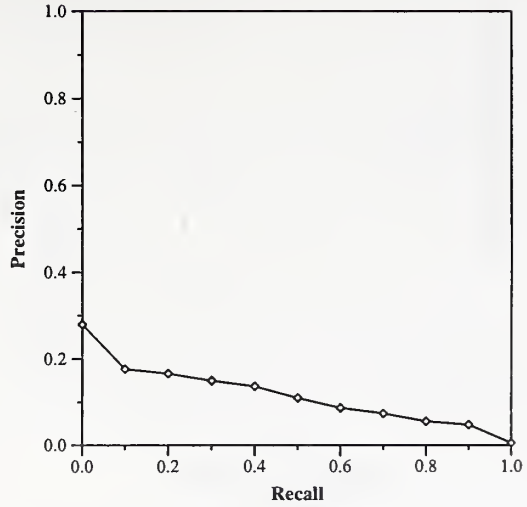
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	603

Recall Level Precision Averages	
Recall	Precision
0.00	0.2797
0.10	0.1769
0.20	0.1664
0.30	0.1499
0.40	0.1367
0.50	0.1099
0.60	0.0869
0.70	0.0741
0.80	0.0557
0.90	0.0480
1.00	0.0055

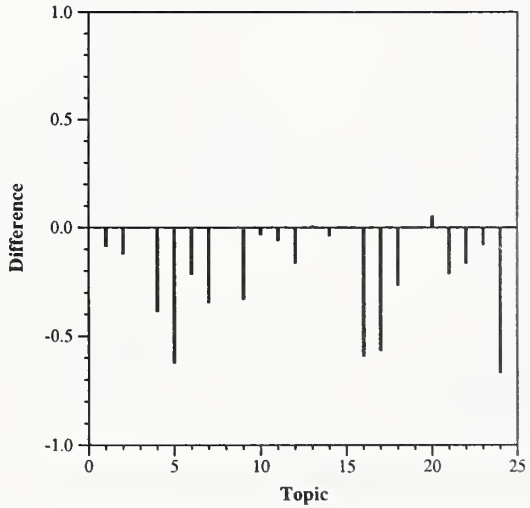
Average precision over all relevant docs	
non-interpolated	0.1095

Document Level Averages	
	Precision
At 5 docs	0.1905
At 10 docs	0.1810
At 15 docs	0.1651
At 20 docs	0.1571
At 30 docs	0.1524
At 100 docs	0.1181
At 200 docs	0.0814
At 500 docs	0.0458
At 1000 docs	0.0287

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1292



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	97lsiLEG
Run Description	Cross-language run: English topics, German documents [unjudged]
Number of Topics	21

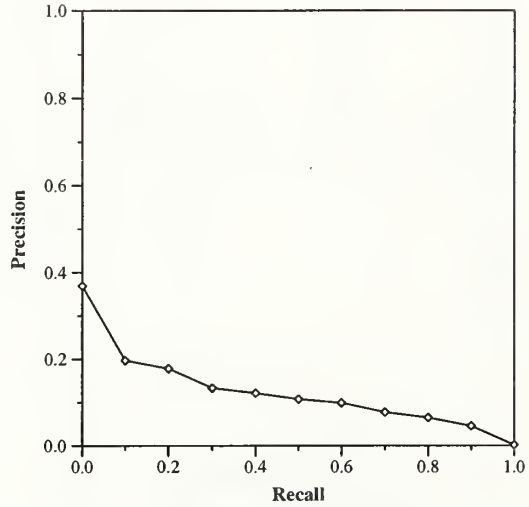
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	425

Recall Level Precision Averages	
Recall	Precision
0.00	0.3694
0.10	0.1978
0.20	0.1788
0.30	0.1340
0.40	0.1219
0.50	0.1080
0.60	0.0993
0.70	0.0778
0.80	0.0650
0.90	0.0456
1.00	0.0018

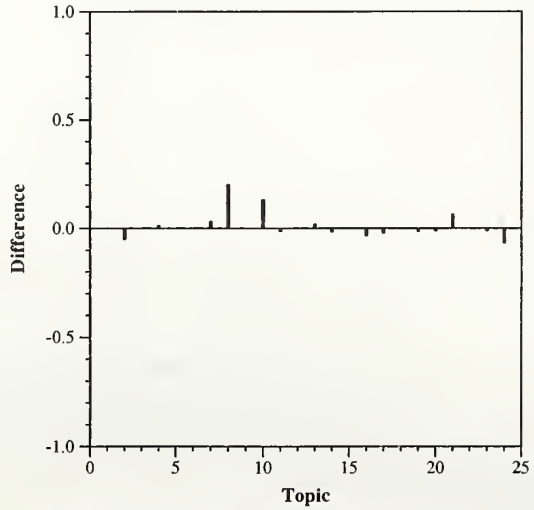
Average precision over all relevant docs	
non-interpolated	0.1144

Document Level Averages	
	Precision
At 5 docs	0.2000
At 10 docs	0.1857
At 15 docs	0.1746
At 20 docs	0.1786
At 30 docs	0.1619
At 100 docs	0.1038
At 200 docs	0.0648
At 500 docs	0.0350
At 1000 docs	0.0202

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1266



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	97lsiLFF
Run Description	Monolingual French run [judged]
Number of Topics	21

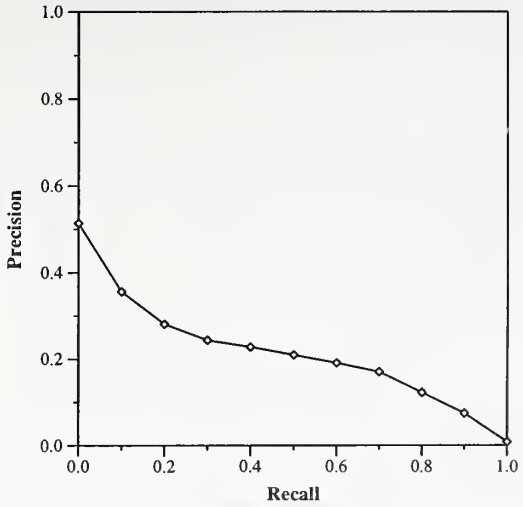
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	856

Recall Level Precision Averages	
Recall	Precision
0.00	0.5141
0.10	0.3561
0.20	0.2814
0.30	0.2441
0.40	0.2288
0.50	0.2101
0.60	0.1915
0.70	0.1709
0.80	0.1231
0.90	0.0746
1.00	0.0083

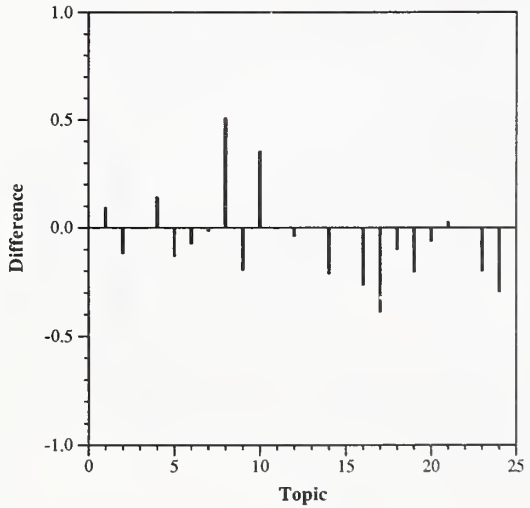
Average precision over all relevant docs	
non-interpolated	0.2011

Document Level Averages	
	Precision
At 5 docs	0.3048
At 10 docs	0.2857
At 15 docs	0.2730
At 20 docs	0.2595
At 30 docs	0.2556
At 100 docs	0.2105
At 200 docs	0.1345
At 500 docs	0.0701
At 1000 docs	0.0408

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2363



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	97lsiLFG
Run Description	Cross-language run: French topics, German documents [unjudged]
Number of Topics	21

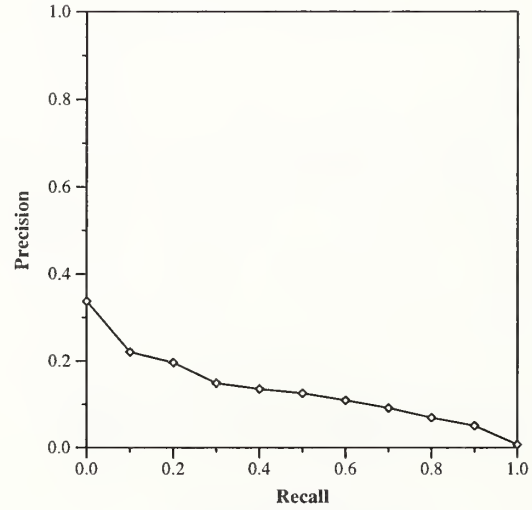
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	600

Recall Level Precision Averages	
Recall	Precision
0.00	0.3377
0.10	0.2209
0.20	0.1966
0.30	0.1491
0.40	0.1357
0.50	0.1259
0.60	0.1095
0.70	0.0920
0.80	0.0695
0.90	0.0510
1.00	0.0070

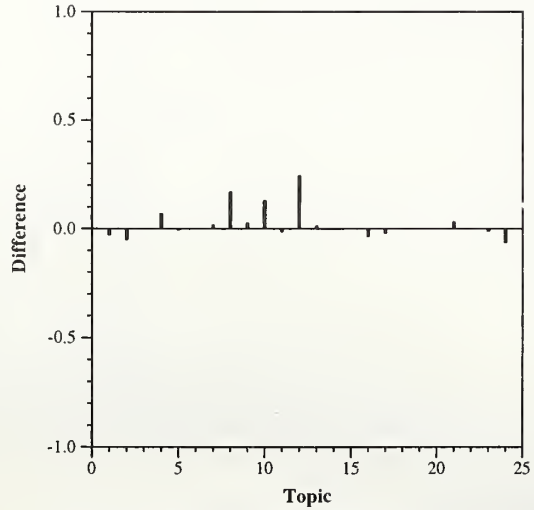
Average precision over all relevant docs	
non-interpolated	0.1263

Document Level Averages	
	Precision
At 5 docs	0.2190
At 10 docs	0.2190
At 15 docs	0.2000
At 20 docs	0.1833
At 30 docs	0.1698
At 100 docs	0.1257
At 200 docs	0.0869
At 500 docs	0.0469
At 1000 docs	0.0286

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1447



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	97lsiLGF
Run Description	Cross-language run: German topics, French documents [unjudged]
Number of Topics	21

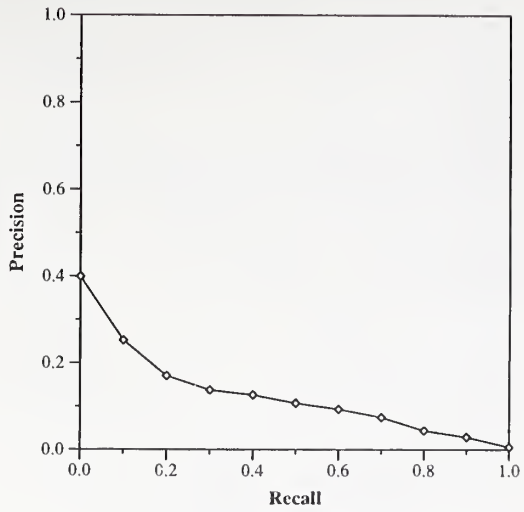
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	699

Recall Level Precision Averages	
Recall	Precision
0.00	0.3994
0.10	0.2527
0.20	0.1706
0.30	0.1379
0.40	0.1264
0.50	0.1075
0.60	0.0935
0.70	0.0748
0.80	0.0441
0.90	0.0292
1.00	0.0060

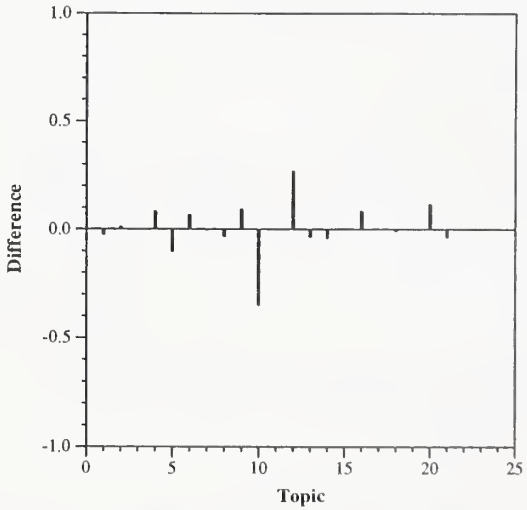
Average precision over all relevant docs	
non-interpolated	0.1152

Document Level Averages	
	Precision
At 5 docs	0.2286
At 10 docs	0.2143
At 15 docs	0.1937
At 20 docs	0.1881
At 30 docs	0.1762
At 100 docs	0.1248
At 200 docs	0.0940
At 500 docs	0.0549
At 1000 docs	0.0333

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1291



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics		
Run Number		97lsiLGG
Run Description	Monolingual German run [judged]	
Number of Topics		21

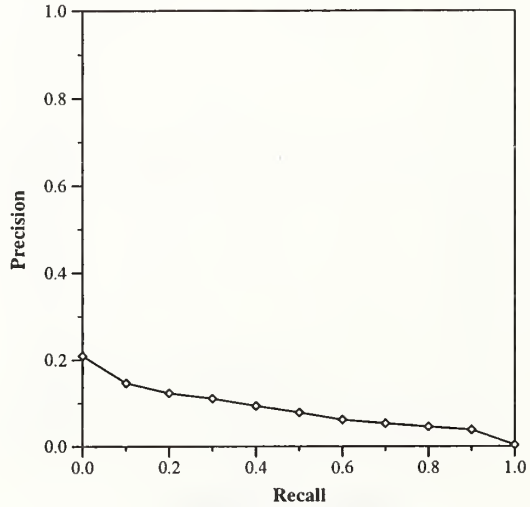
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	423

Recall Level Precision Averages	
Recall	Precision
0.00	0.2093
0.10	0.1468
0.20	0.1233
0.30	0.1108
0.40	0.0940
0.50	0.0787
0.60	0.0618
0.70	0.0534
0.80	0.0456
0.90	0.0386
1.00	0.0035

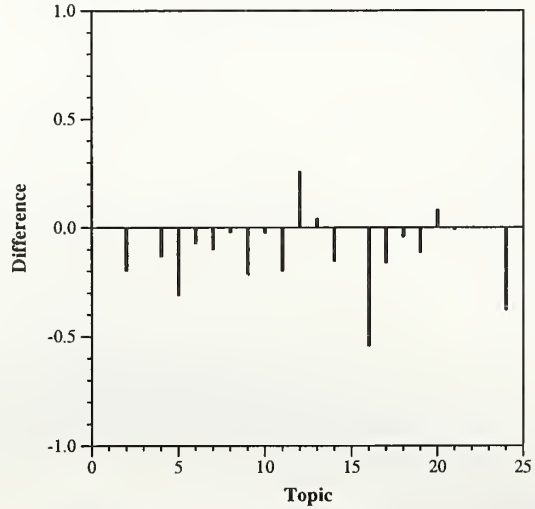
Average precision over all relevant docs	
non-interpolated	0.0807

Document Level Averages	
	Precision
At 5 docs	0.1143
At 10 docs	0.1429
At 15 docs	0.1238
At 20 docs	0.1238
At 30 docs	0.1270
At 100 docs	0.1033
At 200 docs	0.0683
At 500 docs	0.0346
At 1000 docs	0.0201

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0935



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	MercureFFl
Run Description	Monolingual French run [judged]
Number of Topics	21

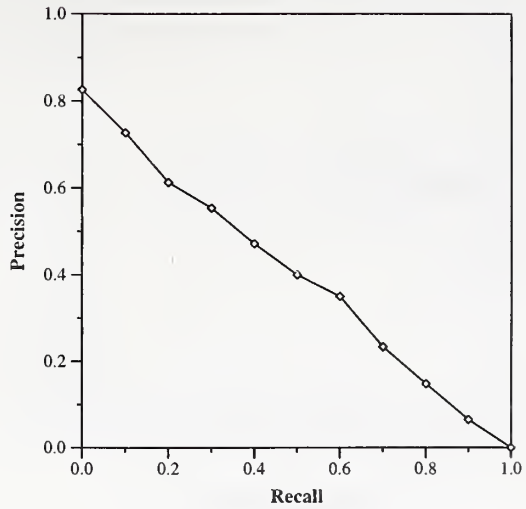
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	1033

Recall Level Precision Averages	
Recall	Precision
0.00	0.8262
0.10	0.7274
0.20	0.6133
0.30	0.5541
0.40	0.4717
0.50	0.3999
0.60	0.3500
0.70	0.2335
0.80	0.1480
0.90	0.0651
1.00	0.0000

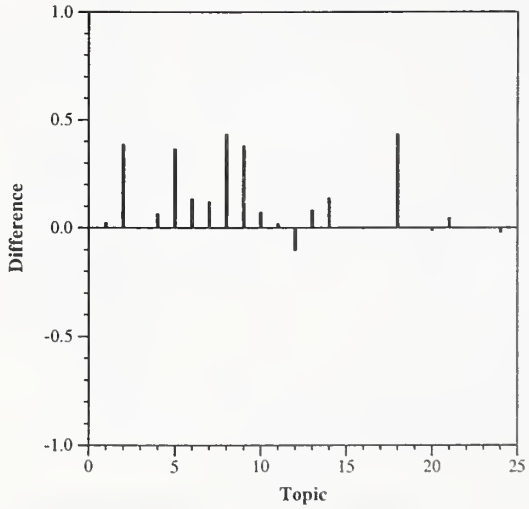
Average precision over all relevant docs	
non-interpolated	0.3778

Document Level Averages	
	Precision
At 5 docs	0.5714
At 10 docs	0.5286
At 15 docs	0.4698
At 20 docs	0.4357
At 30 docs	0.3889
At 100 docs	0.2767
At 200 docs	0.1883
At 500 docs	0.0912
At 1000 docs	0.0492

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4015



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	clcrl1
Run Description	Monolingual French run [judged]
Number of Topics	21

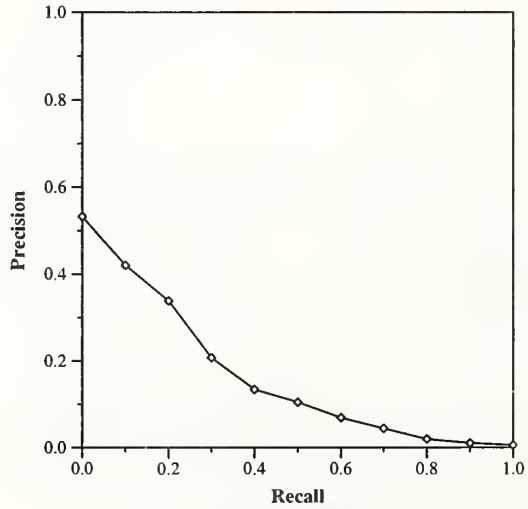
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	758

Recall Level Precision Averages	
Recall	Precision
0.00	0.5327
0.10	0.4208
0.20	0.3389
0.30	0.2078
0.40	0.1347
0.50	0.1052
0.60	0.0691
0.70	0.0443
0.80	0.0196
0.90	0.0104
1.00	0.0057

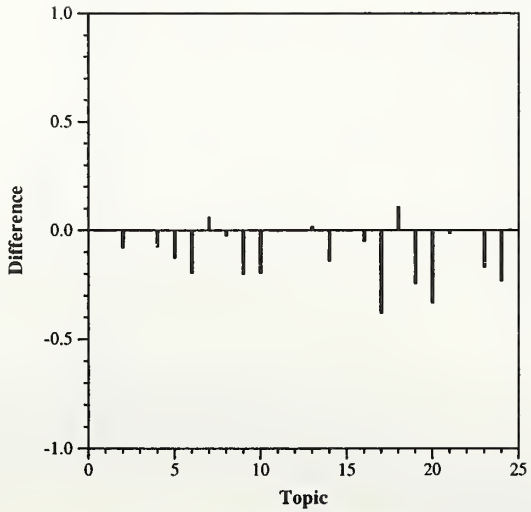
Average precision over all relevant docs	
non-interpolated	0.1484

Document Level Averages	
	Precision
At 5 docs	0.3333
At 10 docs	0.2762
At 15 docs	0.2571
At 20 docs	0.2452
At 30 docs	0.2222
At 100 docs	0.1510
At 200 docs	0.1045
At 500 docs	0.0585
At 1000 docs	0.0361

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1947



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	clcrl3
Run Description	Cross-language run: English topics, French documents [unjudged]
Number of Topics	21

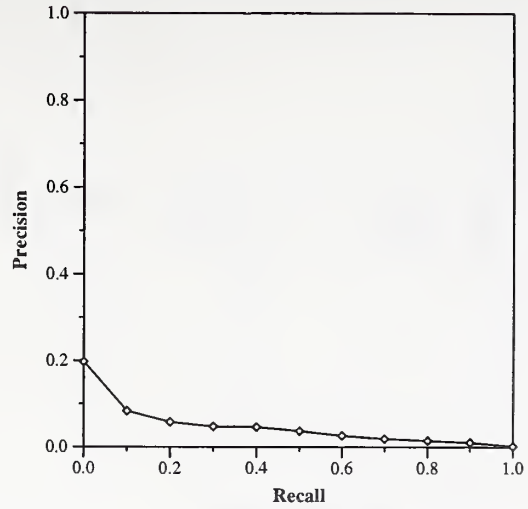
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	308

Recall Level Precision Averages	
Recall	Precision
0.00	0.1976
0.10	0.0838
0.20	0.0582
0.30	0.0477
0.40	0.0466
0.50	0.0372
0.60	0.0263
0.70	0.0193
0.80	0.0150
0.90	0.0106
1.00	0.0021

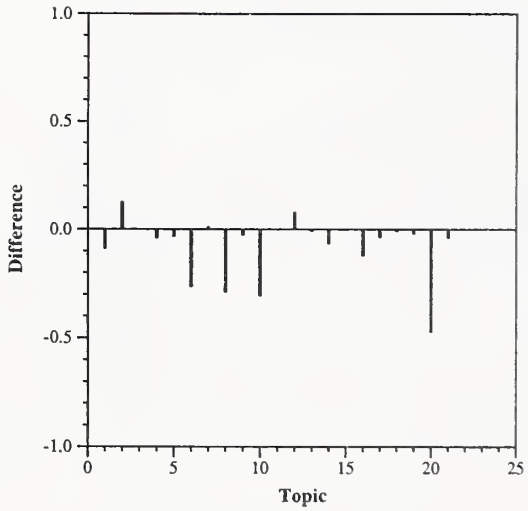
Average precision over all relevant docs	
non-interpolated	0.0357

Document Level Averages	
	Precision
At 5 docs	0.1143
At 10 docs	0.1000
At 15 docs	0.0921
At 20 docs	0.0786
At 30 docs	0.0619
At 100 docs	0.0505
At 200 docs	0.0343
At 500 docs	0.0217
At 1000 docs	0.0147

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0566



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ETHdd1
Run Description	Monolingual German [judged] run
Number of Topics	21

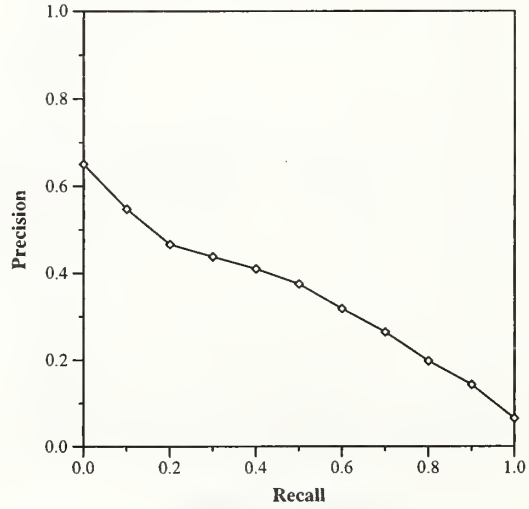
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	865

Recall Level Precision Averages	
Recall	Precision
0.00	0.6505
0.10	0.5476
0.20	0.4665
0.30	0.4380
0.40	0.4103
0.50	0.3752
0.60	0.3184
0.70	0.2644
0.80	0.1977
0.90	0.1429
1.00	0.0650

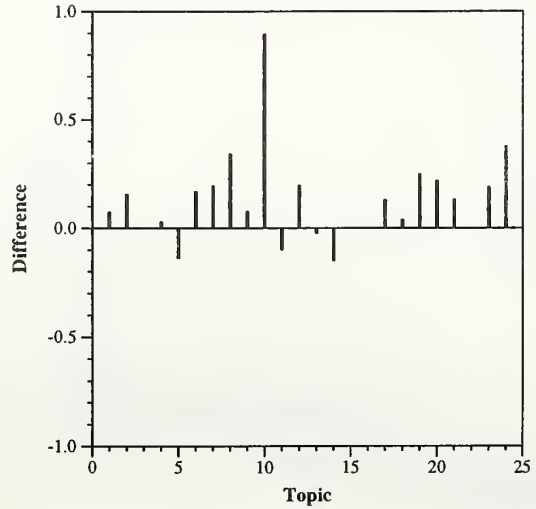
Average precision over all relevant docs	
non-interpolated	0.3349

Document Level Averages	
	Precision
At 5 docs	0.4476
At 10 docs	0.4429
At 15 docs	0.4222
At 20 docs	0.3905
At 30 docs	0.3651
At 100 docs	0.2610
At 200 docs	0.1533
At 500 docs	0.0747
At 1000 docs	0.0412

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3592



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ETHde1
Run Description	Cross-language run: German topics, English documents [unjudged]
Number of Topics	21

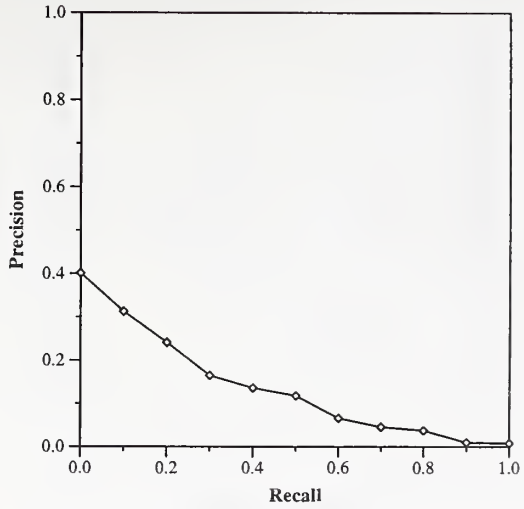
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	689

Recall Level Precision Averages	
Recall	Precision
0.00	0.4011
0.10	0.3129
0.20	0.2416
0.30	0.1653
0.40	0.1364
0.50	0.1184
0.60	0.0665
0.70	0.0460
0.80	0.0378
0.90	0.0100
1.00	0.0083

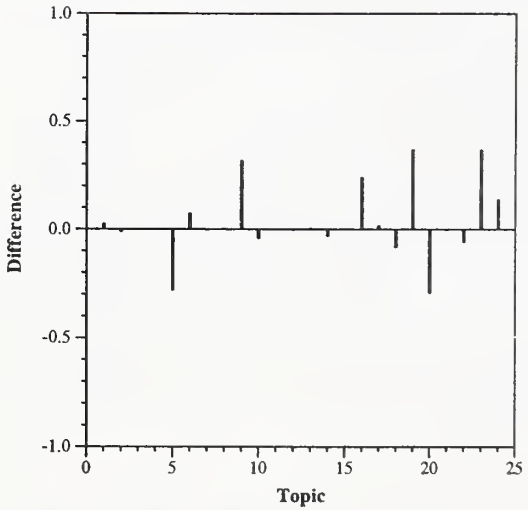
Average precision over all relevant docs	
non-interpolated	0.1293

Document Level Averages	
	Precision
At 5 docs	0.3048
At 10 docs	0.2476
At 15 docs	0.2190
At 20 docs	0.1857
At 30 docs	0.1603
At 100 docs	0.0986
At 200 docs	0.0738
At 500 docs	0.0484
At 1000 docs	0.0328

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1303



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ETHed1
Run Description	Cross-language run: English topics, German documents [unjudged]
Number of Topics	21

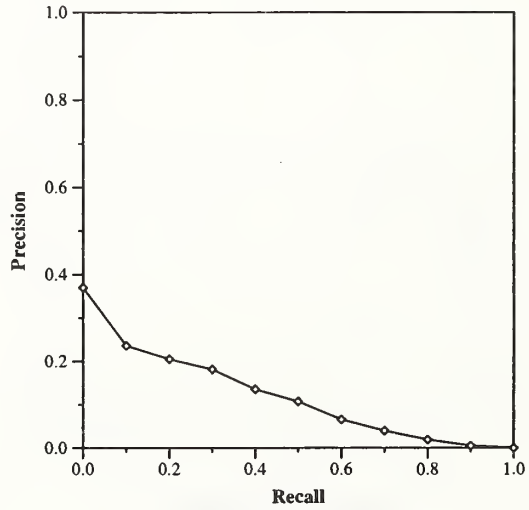
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	495

Recall Level Precision Averages	
Recall	Precision
0.00	0.3699
0.10	0.2363
0.20	0.2055
0.30	0.1817
0.40	0.1358
0.50	0.1075
0.60	0.0659
0.70	0.0399
0.80	0.0193
0.90	0.0052
1.00	0.0000

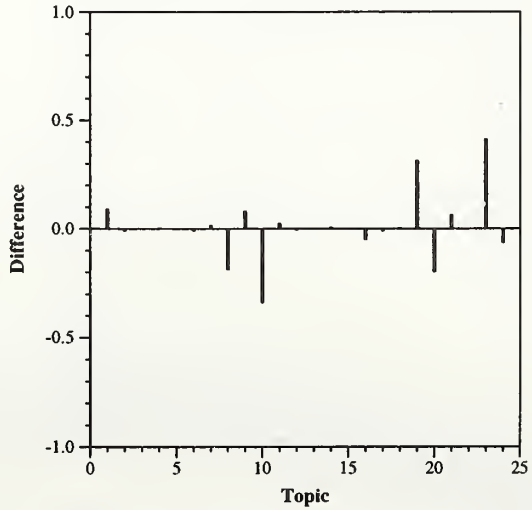
Average precision over all relevant docs	
non-interpolated	0.1104

Document Level Averages	
	Precision
At 5 docs	0.2381
At 10 docs	0.2190
At 15 docs	0.1905
At 20 docs	0.1667
At 30 docs	0.1429
At 100 docs	0.0943
At 200 docs	0.0660
At 500 docs	0.0386
At 1000 docs	0.0236

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1494



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ETHee1
Run Description	Monolingual English run [judged]
Number of Topics	21

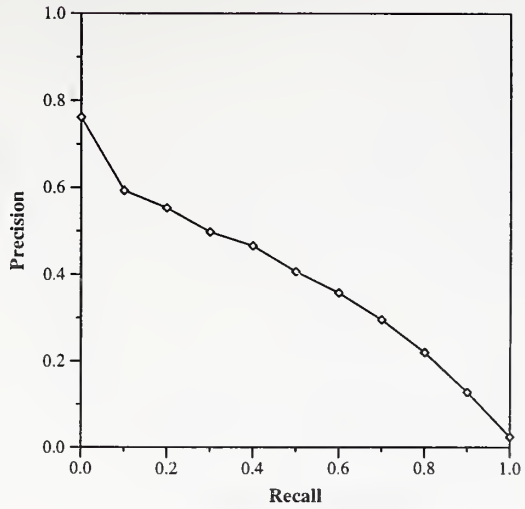
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	1165

Recall Level Precision Averages	
Recall	Precision
0.00	0.7617
0.10	0.5934
0.20	0.5530
0.30	0.4978
0.40	0.4657
0.50	0.4064
0.60	0.3574
0.70	0.2954
0.80	0.2203
0.90	0.1281
1.00	0.0241

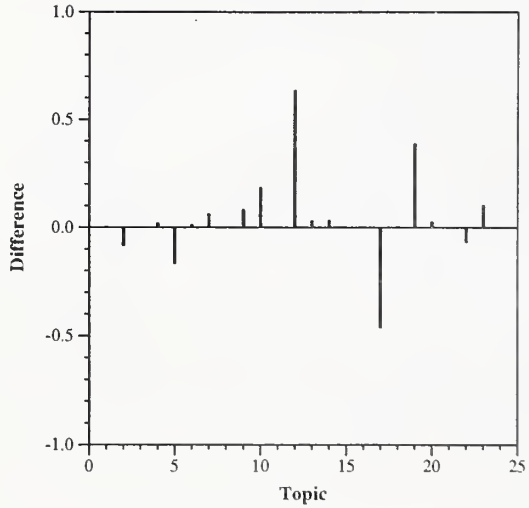
Average precision over all relevant docs	
non-interpolated	0.3793

Document Level Averages	
	Precision
At 5 docs	0.5238
At 10 docs	0.5238
At 15 docs	0.5048
At 20 docs	0.4690
At 30 docs	0.4286
At 100 docs	0.3167
At 200 docs	0.2036
At 500 docs	0.1021
At 1000 docs	0.0555

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3694



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ETHfd1
Run Description	Cross-language run: French topics, German documents [unjudged]
Number of Topics	21

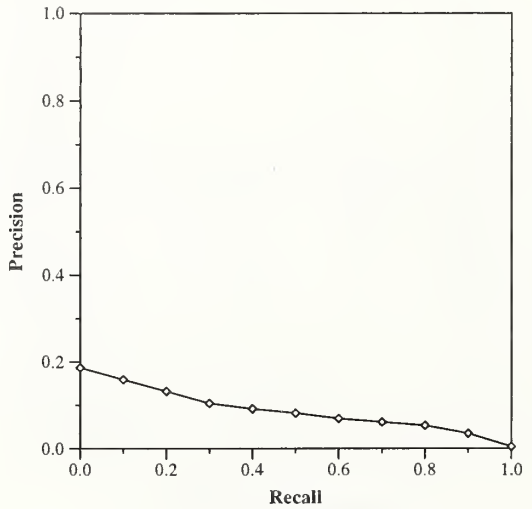
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	421

Recall Level Precision Averages	
Recall	Precision
0.00	0.1871
0.10	0.1599
0.20	0.1328
0.30	0.1049
0.40	0.0921
0.50	0.0822
0.60	0.0696
0.70	0.0615
0.80	0.0535
0.90	0.0350
1.00	0.0045

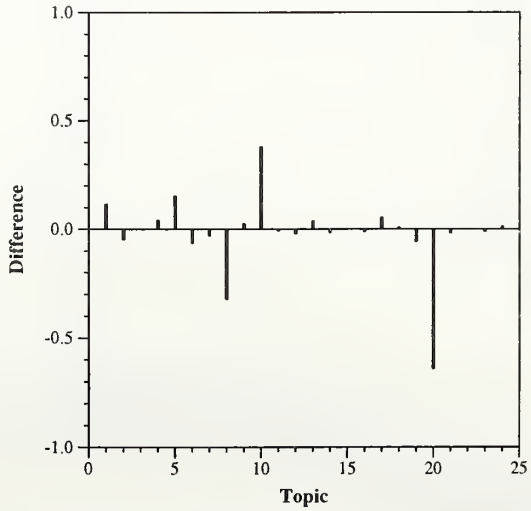
Average precision over all relevant docs	
non-interpolated	0.0840

Document Level Averages	
	Precision
At 5 docs	0.0857
At 10 docs	0.1143
At 15 docs	0.1079
At 20 docs	0.0976
At 30 docs	0.1000
At 100 docs	0.0738
At 200 docs	0.0529
At 500 docs	0.0316
At 1000 docs	0.0200

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0969



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	ETHff1
Run Description	Monolingual French run [judged]
Number of Topics	21

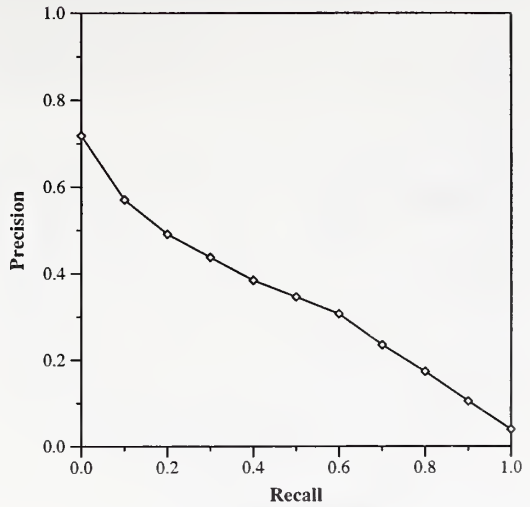
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	1108

Recall Level Precision Averages	
Recall	Precision
0.00	0.7185
0.10	0.5707
0.20	0.4917
0.30	0.4381
0.40	0.3847
0.50	0.3463
0.60	0.3071
0.70	0.2355
0.80	0.1737
0.90	0.1052
1.00	0.0396

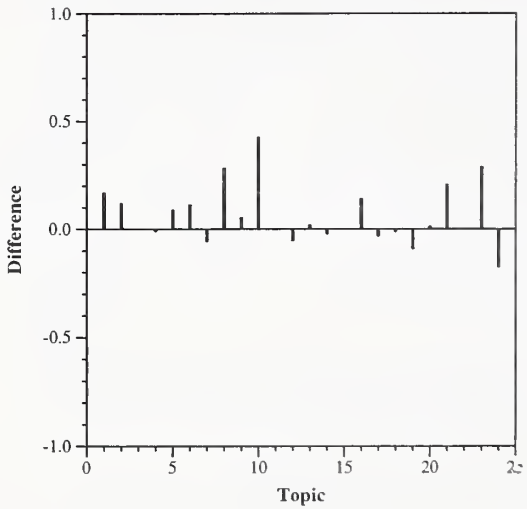
Average precision over all relevant docs	
non-interpolated	0.3261

Document Level Averages	
	Precision
At 5 docs	0.4857
At 10 docs	0.4714
At 15 docs	0.4127
At 20 docs	0.3786
At 30 docs	0.3540
At 100 docs	0.2724
At 200 docs	0.1790
At 500 docs	0.0937
At 1000 docs	0.0528

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3552



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	TNOde1
Run Description	Cross-language run: German topics, English documents [unjudged]
Number of Topics	21

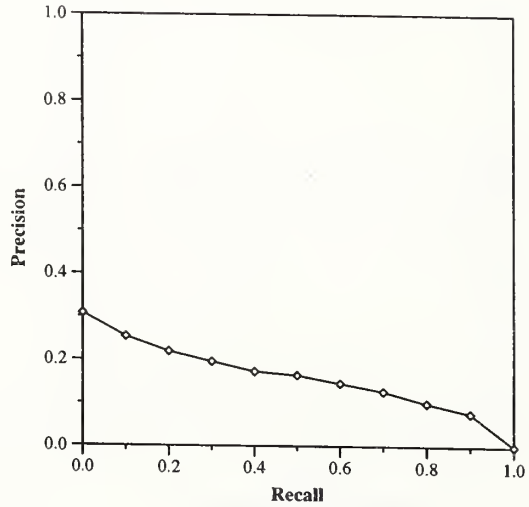
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	657

Recall Level Precision Averages	
Recall	Precision
0.00	0.3077
0.10	0.2542
0.20	0.2197
0.30	0.1962
0.40	0.1734
0.50	0.1650
0.60	0.1464
0.70	0.1281
0.80	0.1005
0.90	0.0778
1.00	0.0010

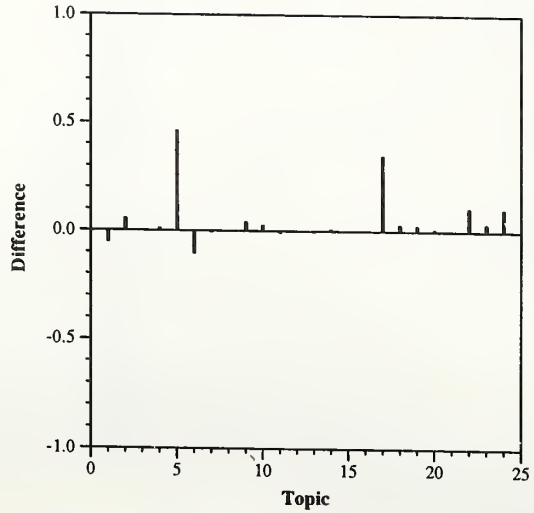
Average precision over all relevant docs	
non-interpolated	0.1453

Document Level Averages	
	Precision
At 5 docs	0.1143
At 10 docs	0.1286
At 15 docs	0.1492
At 20 docs	0.1738
At 30 docs	0.1714
At 100 docs	0.1362
At 200 docs	0.0933
At 500 docs	0.0499
At 1000 docs	0.0313

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1925



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	TNOee
Run Description	Monolingual English run [judged]
Number of Topics	21

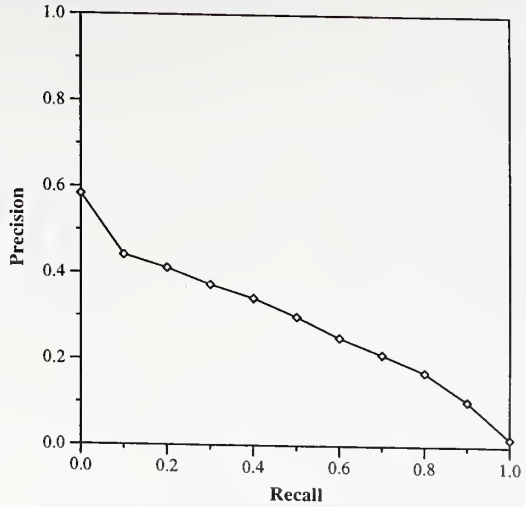
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	1049

Recall Level Precision Averages	
Recall	Precision
0.00	0.5837
0.10	0.4422
0.20	0.4121
0.30	0.3740
0.40	0.3431
0.50	0.3007
0.60	0.2522
0.70	0.2135
0.80	0.1724
0.90	0.1054
1.00	0.0181

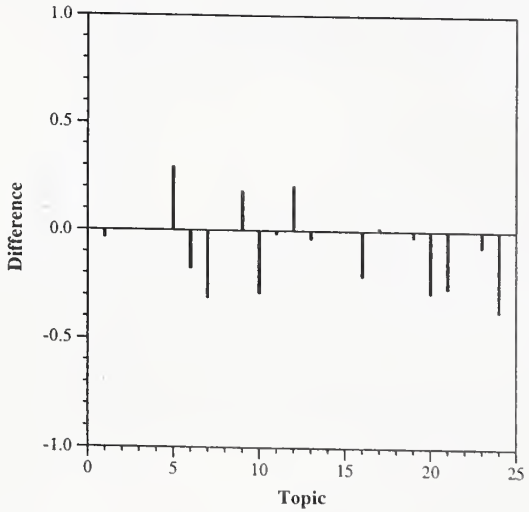
Average precision over all relevant docs	
non-interpolated	0.2752

Document Level Averages	
	Precision
At 5 docs	0.3714
At 10 docs	0.3714
At 15 docs	0.3714
At 20 docs	0.3667
At 30 docs	0.3651
At 100 docs	0.2510
At 200 docs	0.1707
At 500 docs	0.0909
At 1000 docs	0.0500

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2978



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics		
Run Number		TNOfe1
Run Description	Cross-language French topics, English documents [unjudged]	run:
Number of Topics		21

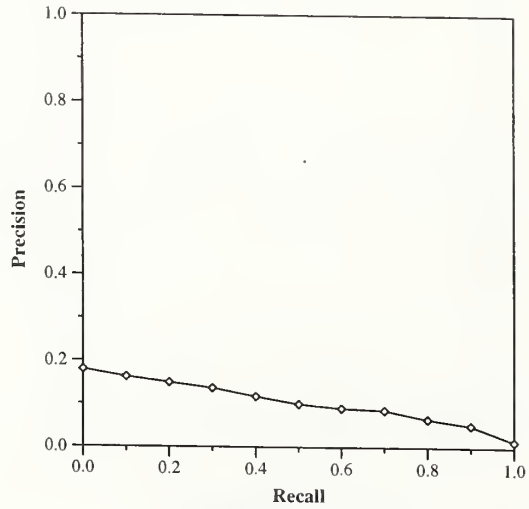
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	584

Recall Level Precision Averages	
Recall	Precision
0.00	0.1795
0.10	0.1625
0.20	0.1497
0.30	0.1368
0.40	0.1178
0.50	0.1003
0.60	0.0910
0.70	0.0866
0.80	0.0660
0.90	0.0515
1.00	0.0148

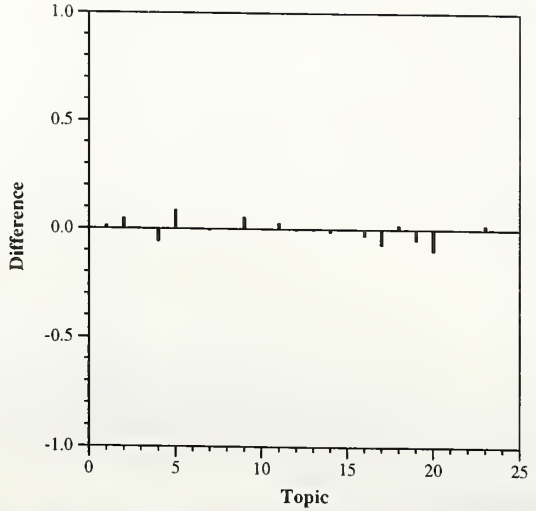
Average precision over all relevant docs	
non-interpolated	0.0913

Document Level Averages	
	Precision
At 5 docs	0.0381
At 10 docs	0.0429
At 15 docs	0.0730
At 20 docs	0.0857
At 30 docs	0.1016
At 100 docs	0.0886
At 200 docs	0.0736
At 500 docs	0.0441
At 1000 docs	0.0278

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1065



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	TNOnle1
Run Description	Cross-language run: Dutch topics, English documents [unjudged]
Number of Topics	21

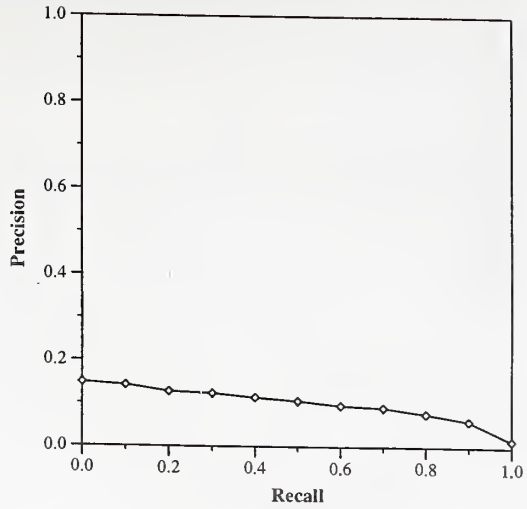
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	626

Recall Level Precision Averages	
Recall	Precision
0.00	0.1484
0.10	0.1423
0.20	0.1278
0.30	0.1239
0.40	0.1144
0.50	0.1068
0.60	0.0964
0.70	0.0917
0.80	0.0778
0.90	0.0615
1.00	0.0159

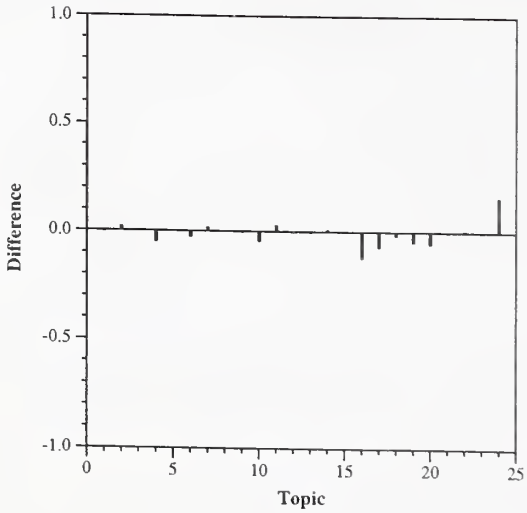
Average precision over all relevant docs	
non-interpolated	0.0841

Document Level Averages	
	Precision
At 5 docs	0.0286
At 10 docs	0.0238
At 15 docs	0.0286
At 20 docs	0.0286
At 30 docs	0.0429
At 100 docs	0.0990
At 200 docs	0.0800
At 500 docs	0.0484
At 1000 docs	0.0298

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0725



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	BrklyE2GA
Run Description	Cross-language run: English topics, German documents [unjudged]
Number of Topics	21

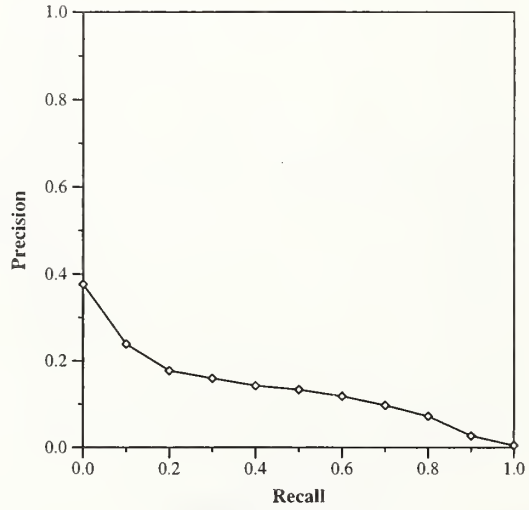
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	295

Recall Level Precision Averages	
Recall	Precision
0.00	0.3766
0.10	0.2390
0.20	0.1778
0.30	0.1598
0.40	0.1429
0.50	0.1337
0.60	0.1186
0.70	0.0977
0.80	0.0724
0.90	0.0274
1.00	0.0048

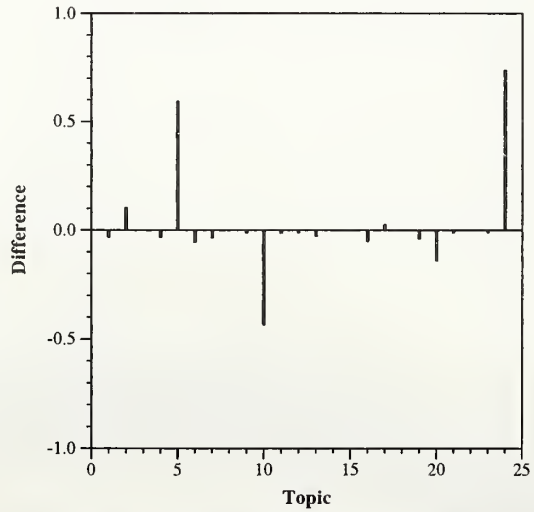
Average precision over all relevant docs	
non-interpolated	0.1305

Document Level Averages	
	Precision
At 5 docs	0.2286
At 10 docs	0.1857
At 15 docs	0.1651
At 20 docs	0.1476
At 30 docs	0.1254
At 100 docs	0.0705
At 200 docs	0.0443
At 500 docs	0.0223
At 1000 docs	0.0140

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.1428



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	BrklyG2GA
Run Description	Monolingual German run [judged]
Number of Topics	21

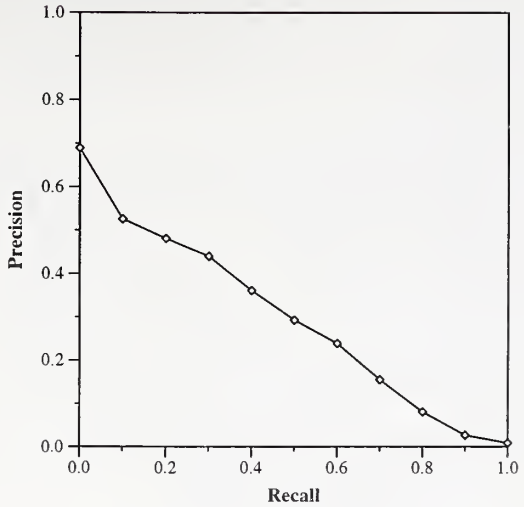
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	675

Recall Level Precision Averages	
Recall	Precision
0.00	0.6894
0.10	0.5259
0.20	0.4809
0.30	0.4402
0.40	0.3614
0.50	0.2933
0.60	0.2391
0.70	0.1556
0.80	0.0810
0.90	0.0269
1.00	0.0088

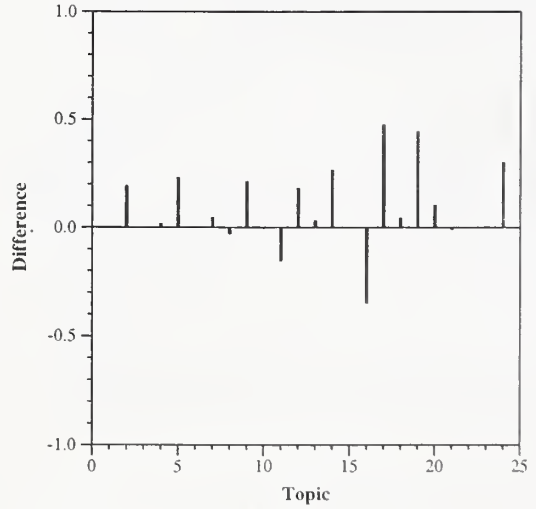
Average precision over all relevant docs	
non-interpolated	0.2845

Document Level Averages	
	Precision
At 5 docs	0.4667
At 10 docs	0.4238
At 15 docs	0.4000
At 20 docs	0.3690
At 30 docs	0.3333
At 100 docs	0.2029
At 200 docs	0.1190
At 500 docs	0.0578
At 1000 docs	0.0321

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3038



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	umcpxeg1
Run Description	Cross-language run: English topics, German documents [unjudged]
Number of Topics	21

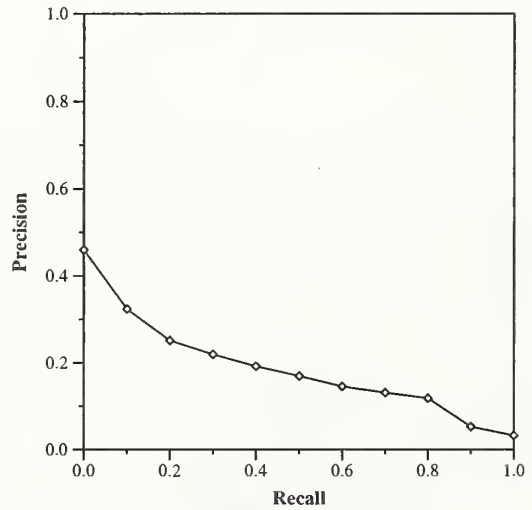
Total number of documents over all topics	
Retrieved:	20001
Relevant:	992
Rel-ret:	576

Recall Level Precision Averages	
Recall	Precision
0.00	0.4601
0.10	0.3241
0.20	0.2519
0.30	0.2198
0.40	0.1923
0.50	0.1697
0.60	0.1457
0.70	0.1314
0.80	0.1186
0.90	0.0528
1.00	0.0325

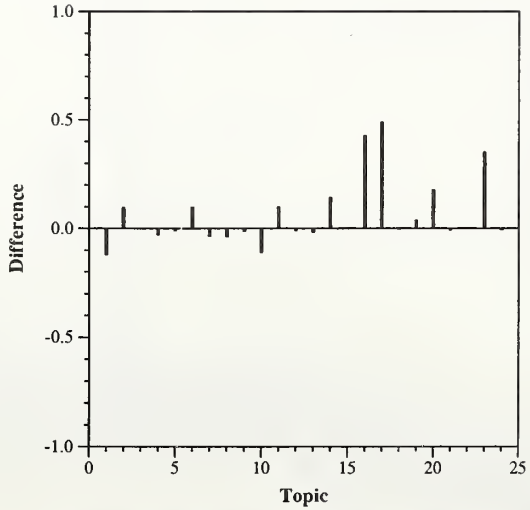
Average precision over all relevant docs	
non-interpolated	0.1761

Document Level Averages	
	Precision
At 5 docs	0.2667
At 10 docs	0.2429
At 15 docs	0.2444
At 20 docs	0.2310
At 30 docs	0.1984
At 100 docs	0.1333
At 200 docs	0.0948
At 500 docs	0.0495
At 1000 docs	0.0274

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2066



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	umcpxge1
Run Description	Cross-language run: German topics, English documents [unjudged]
Number of Topics	21

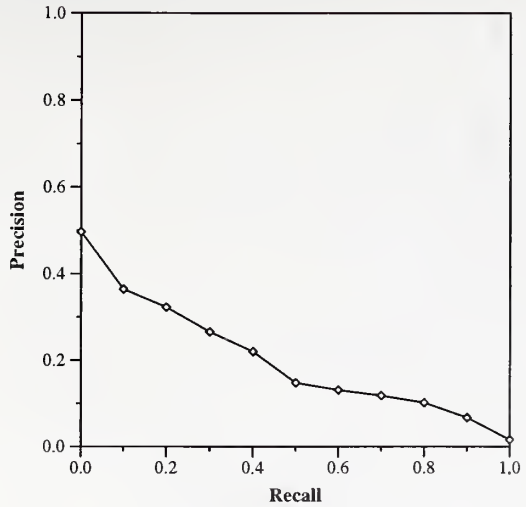
Total number of documents over all topics	
Retrieved:	19002
Relevant:	1247
Rel-ret:	631

Recall Level Precision Averages	
Recall	Precision
0.00	0.4967
0.10	0.3646
0.20	0.3230
0.30	0.2660
0.40	0.2202
0.50	0.1480
0.60	0.1313
0.70	0.1183
0.80	0.1019
0.90	0.0675
1.00	0.0161

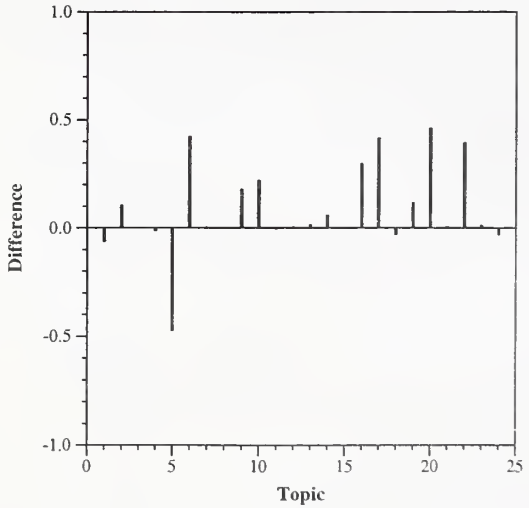
Average precision over all relevant docs	
non-interpolated	0.1928

Document Level Averages	
	Precision
At 5 docs	0.3333
At 10 docs	0.3143
At 15 docs	0.2857
At 20 docs	0.2738
At 30 docs	0.2556
At 100 docs	0.1481
At 200 docs	0.0988
At 500 docs	0.0527
At 1000 docs	0.0300

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2205



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	umcpxgg3
Run Description	Monolingual German run [judged]
Number of Topics	21

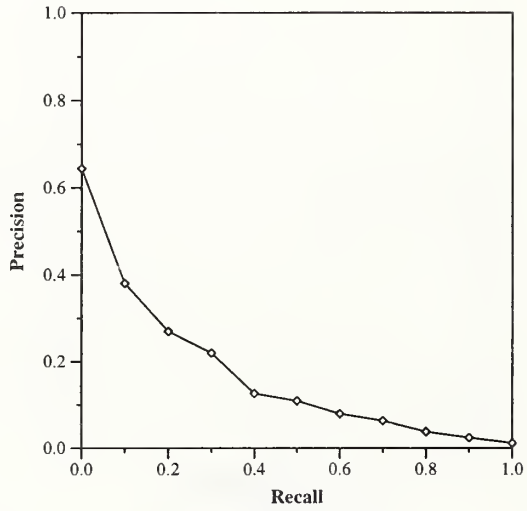
Total number of documents over all topics	
Retrieved:	21000
Relevant:	992
Rel-ret:	520

Recall Level Precision Averages	
Recall	Precision
0.00	0.6445
0.10	0.3809
0.20	0.2698
0.30	0.2199
0.40	0.1263
0.50	0.1094
0.60	0.0795
0.70	0.0635
0.80	0.0380
0.90	0.0244
1.00	0.0117

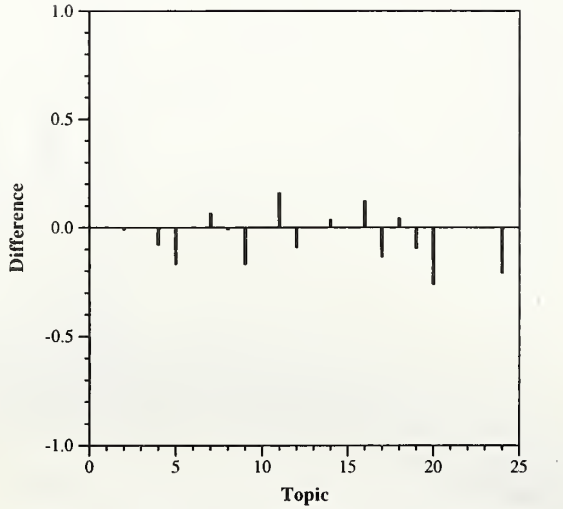
Average precision over all relevant docs	
non-interpolated	0.1519

Document Level Averages	
	Precision
At 5 docs	0.3619
At 10 docs	0.2667
At 15 docs	0.2190
At 20 docs	0.1952
At 30 docs	0.1746
At 100 docs	0.1095
At 200 docs	0.0724
At 500 docs	0.0403
At 1000 docs	0.0248

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2012



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	INQ4xl1
Run Description	Cross-language run: Spanish topics, English documents [judged]
Number of Topics	21

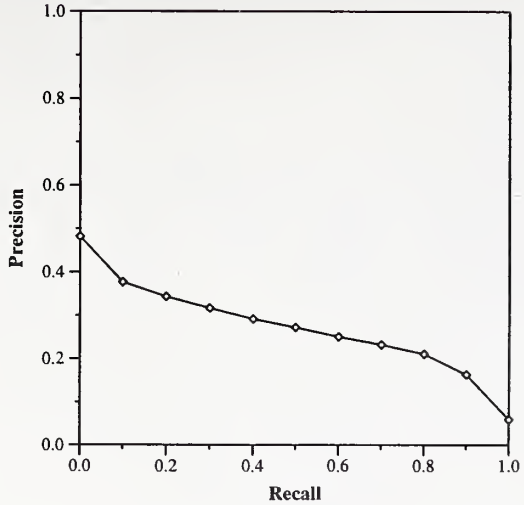
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	757

Recall Level Precision Averages	
Recall	Precision
0.00	0.4822
0.10	0.3769
0.20	0.3433
0.30	0.3168
0.40	0.2919
0.50	0.2724
0.60	0.2508
0.70	0.2326
0.80	0.2110
0.90	0.1633
1.00	0.0592

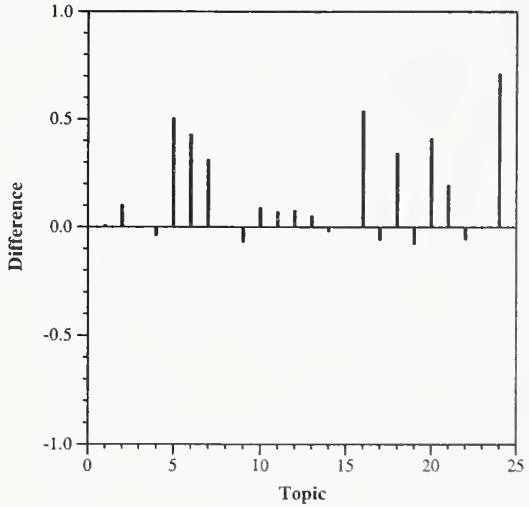
Average precision over all relevant docs	
non-interpolated	0.2610

Document Level Averages	
	Precision
At 5 docs	0.3048
At 10 docs	0.3333
At 15 docs	0.3143
At 20 docs	0.3048
At 30 docs	0.2778
At 100 docs	0.2010
At 200 docs	0.1310
At 500 docs	0.0664
At 1000 docs	0.0360

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2767



Recall-Precision Curve

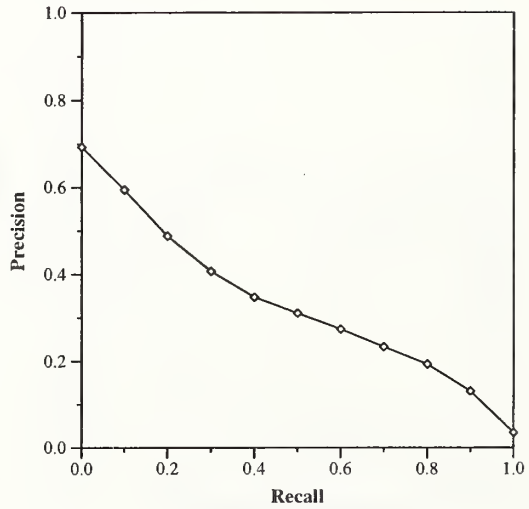


Difference from Median in Average Precision per Topic

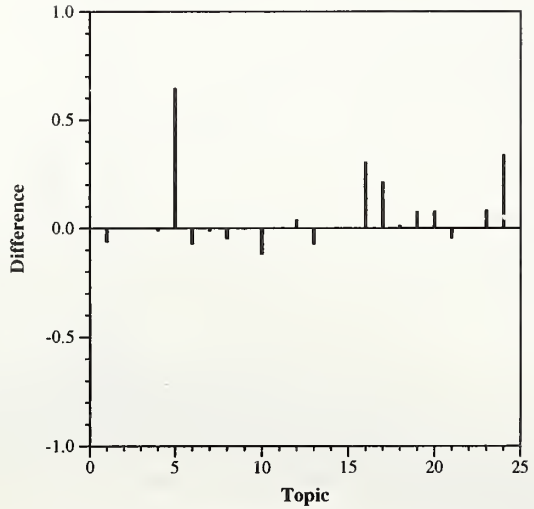
Summary Statistics	
Run Number	CLIPS1
Run Description	Monolingual French run [judged]
Number of Topics	21
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	989

Recall Level Precision Averages	
Recall	Precision
0.00	0.6932
0.10	0.5949
0.20	0.4887
0.30	0.4072
0.40	0.3481
0.50	0.3109
0.60	0.2741
0.70	0.2331
0.80	0.1930
0.90	0.1307
1.00	0.0350
Average precision over all relevant docs	
non-interpolated	0.3204

Document Level Averages	
	Precision
At 5 docs	0.5238
At 10 docs	0.4714
At 15 docs	0.4159
At 20 docs	0.3857
At 30 docs	0.3524
At 100 docs	0.2229
At 200 docs	0.1505
At 500 docs	0.0783
At 1000 docs	0.0471
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3439



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	XRCECLE2EM
Run Description	Monolingual English run [judged]
Number of Topics	21

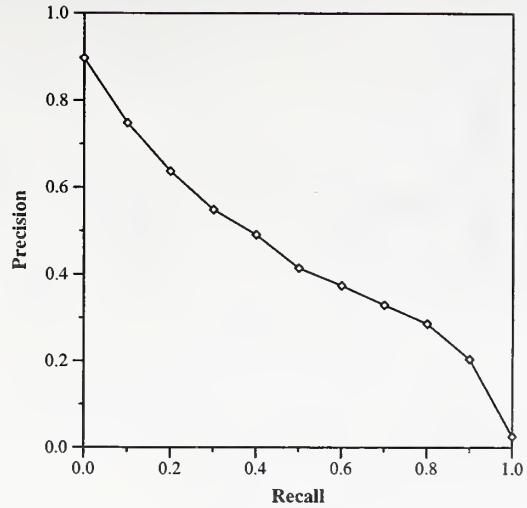
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	1006

Recall Level Precision Averages	
Recall	Precision
0.00	0.8967
0.10	0.7486
0.20	0.6377
0.30	0.5491
0.40	0.4914
0.50	0.4144
0.60	0.3742
0.70	0.3294
0.80	0.2862
0.90	0.2046
1.00	0.0255

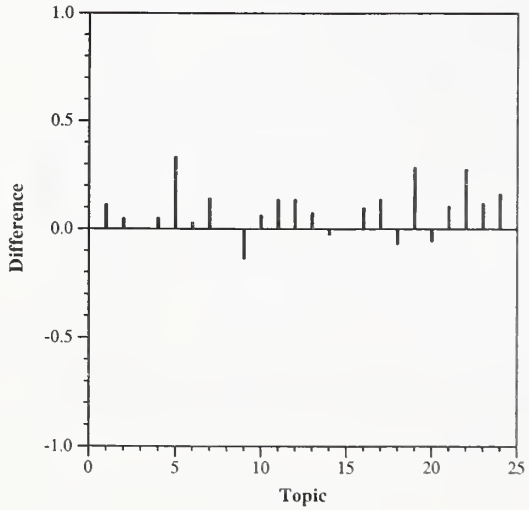
Average precision over all relevant docs	
non-interpolated	0.4375

Document Level Averages	
	Precision
At 5 docs	0.6667
At 10 docs	0.5952
At 15 docs	0.5778
At 20 docs	0.5405
At 30 docs	0.4905
At 100 docs	0.3390
At 200 docs	0.2010
At 500 docs	0.0914
At 1000 docs	0.0479

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4349



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	XRCECLE2FM
Run Description	Cross-language run: English topics, French documents [unjudged]
Number of Topics	21

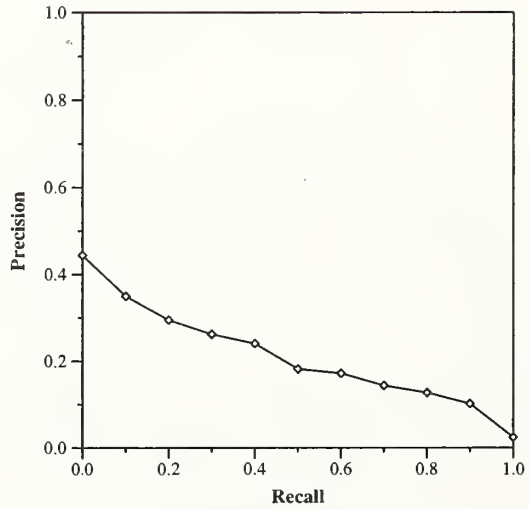
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	637

Recall Level Precision Averages	
Recall	Precision
0.00	0.4441
0.10	0.3500
0.20	0.2952
0.30	0.2625
0.40	0.2411
0.50	0.1824
0.60	0.1725
0.70	0.1442
0.80	0.1276
0.90	0.1019
1.00	0.0243

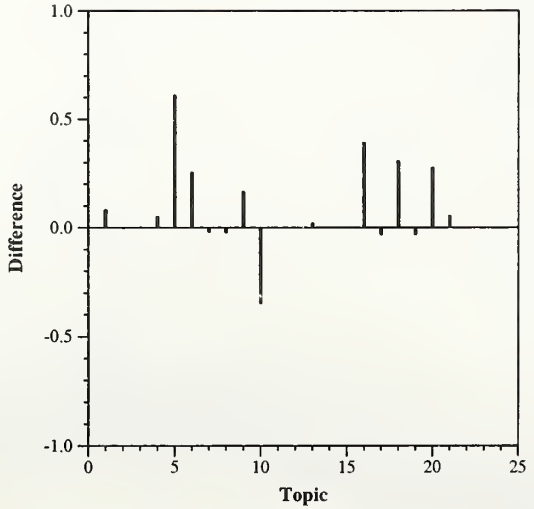
Average precision over all relevant docs	
non-interpolated	0.1946

Document Level Averages	
	Precision
At 5 docs	0.2952
At 10 docs	0.2952
At 15 docs	0.2730
At 20 docs	0.2500
At 30 docs	0.2206
At 100 docs	0.1495
At 200 docs	0.1002
At 500 docs	0.0537
At 1000 docs	0.0303

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2196



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics		
Run Number		XRCECLF2EM
Run Description	Cross-language French topics, English documents [unjudged]	run:
Number of Topics		21

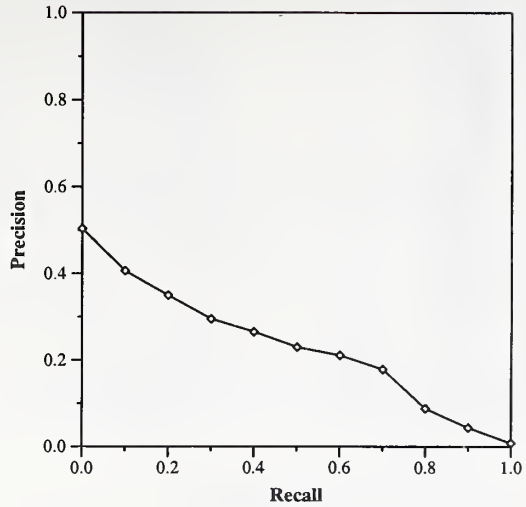
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1247
Rel-ret:	660

Recall Level Precision Averages	
Recall	Precision
0.00	0.5037
0.10	0.4063
0.20	0.3501
0.30	0.2955
0.40	0.2659
0.50	0.2305
0.60	0.2115
0.70	0.1791
0.80	0.0883
0.90	0.0438
1.00	0.0078

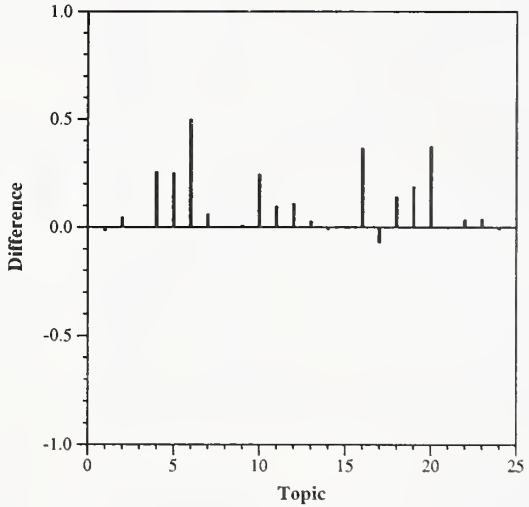
Average precision over all relevant docs	
non-interpolated	0.2183

Document Level Averages	
	Precision
At 5 docs	0.3429
At 10 docs	0.3476
At 15 docs	0.3270
At 20 docs	0.3119
At 30 docs	0.2937
At 100 docs	0.1829
At 200 docs	0.1195
At 500 docs	0.0582
At 1000 docs	0.0314

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2511



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	XRCECLF2FM
Run Description	Monolingual French run [judged]
Number of Topics	21

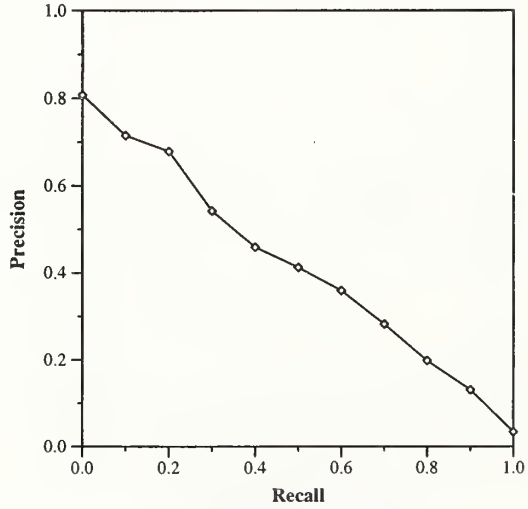
Total number of documents over all topics	
Retrieved:	21000
Relevant:	1239
Rel-ret:	960

Recall Level Precision Averages	
Recall	Precision
0.00	0.8079
0.10	0.7153
0.20	0.6790
0.30	0.5421
0.40	0.4591
0.50	0.4128
0.60	0.3595
0.70	0.2824
0.80	0.1982
0.90	0.1311
1.00	0.0340

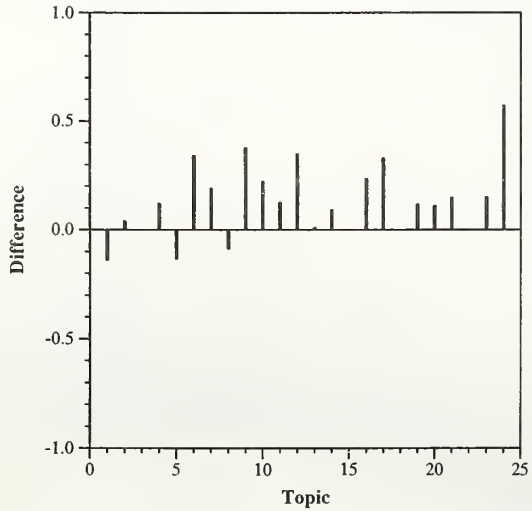
Average precision over all relevant docs	
non-interpolated	0.4073

Document Level Averages	
	Precision
At 5 docs	0.6286
At 10 docs	0.5714
At 15 docs	0.5333
At 20 docs	0.5167
At 30 docs	0.4683
At 100 docs	0.3110
At 200 docs	0.1898
At 500 docs	0.0870
At 1000 docs	0.0457

R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.4412



Recall-Precision Curve



Difference from Median in Average Precision per Topic

Summary Statistics	
Run Number	Cor6HP1
Number of Topics	50
Total number of documents over all topics	
Retrieved:	500
Relevant:	4611
Rel-ret:	272

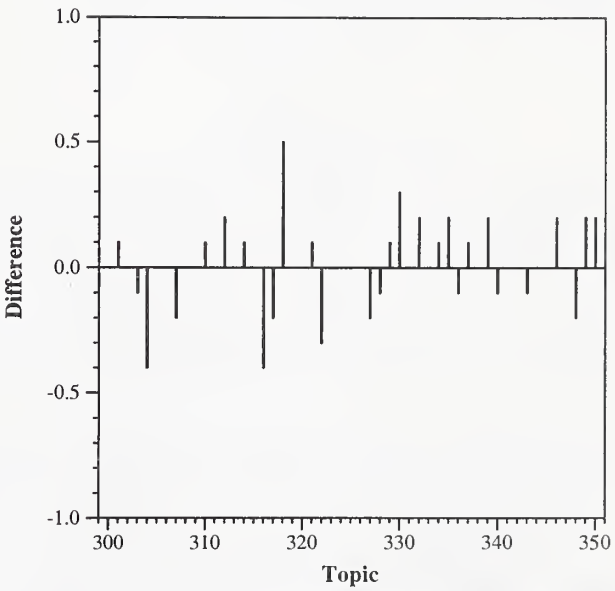
Means over 50 topics	
Precision at 10 Docs:	0.5440
Relative-Precision at 10 Docs:	0.5564
Unranked-Avg-Precision at 10 Docs:	0.0799

Evaluation Measures

Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

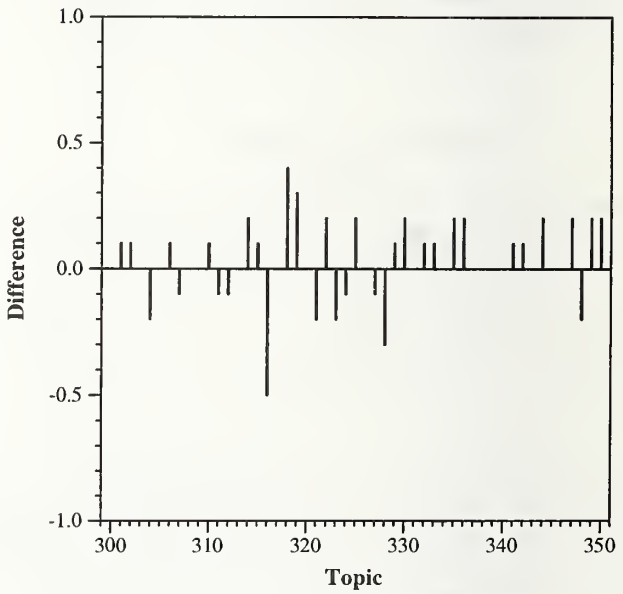
Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.



Precision(10) Difference from Median per Topic

Summary Statistics	
Run Number	Cor6HP2
Number of Topics	50
Total number of documents over all topics	
Retrieved:	500
Relevant:	4611
Rel-ret:	283

Means over 50 topics	
Precision at 10 Docs:	0.5660
Relative-Precision at 10 Docs:	0.5820
Unranked-Avg-Precision at 10 Docs:	0.0786



Precision(10) Difference from Median per Topic

Evaluation Measures

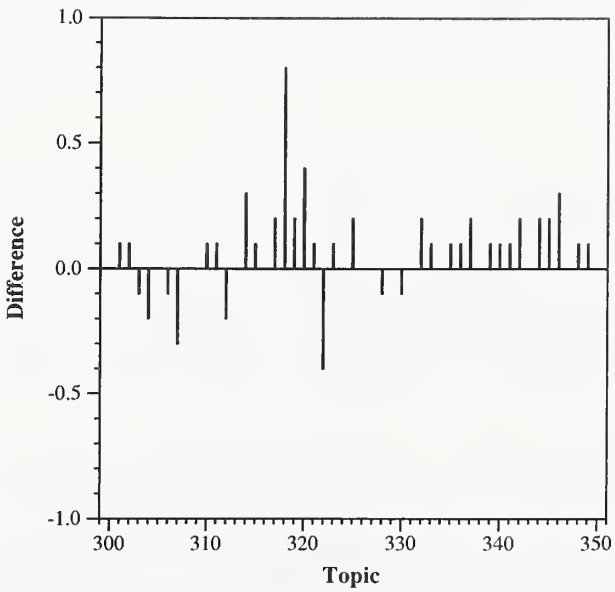
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	Cor6HP3
Number of Topics	50
Total number of documents over all topics	
Retrieved:	500
Relevant:	4611
Rel-ret:	301

Means over 50 topics	
Precision at 10 Docs:	0.6020
Relative-Precision at 10 Docs:	0.6298
Unranked-Avg-Precision at 10 Docs:	0.1021



Precision(10) Difference from Median per Topic

Evaluation Measures

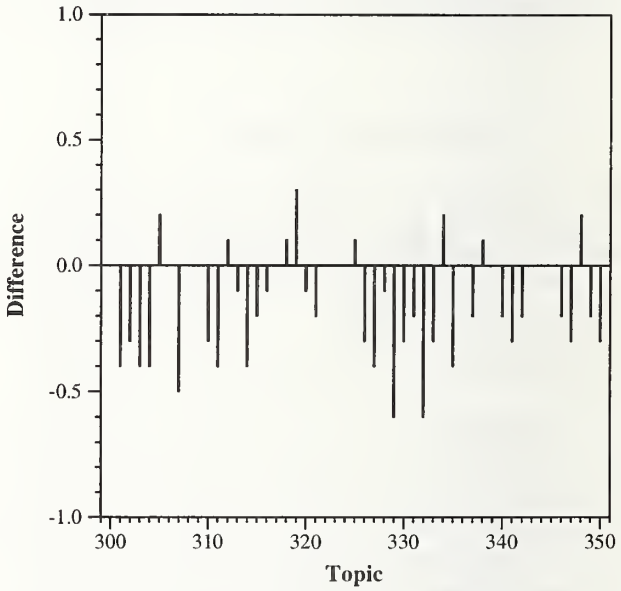
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	DCU97HP
Number of Topics	50
Total number of documents over all topics	
Retrieved:	500
Relevant:	4611
Rel-ret:	191

Means over 50 topics	
Precision at 10 Docs:	0.3820
Relative-Precision at 10 Docs:	0.4031
Unranked-Avg-Precision at 10 Docs:	0.0633



Precision(10) Difference from Median per Topic

Evaluation Measures

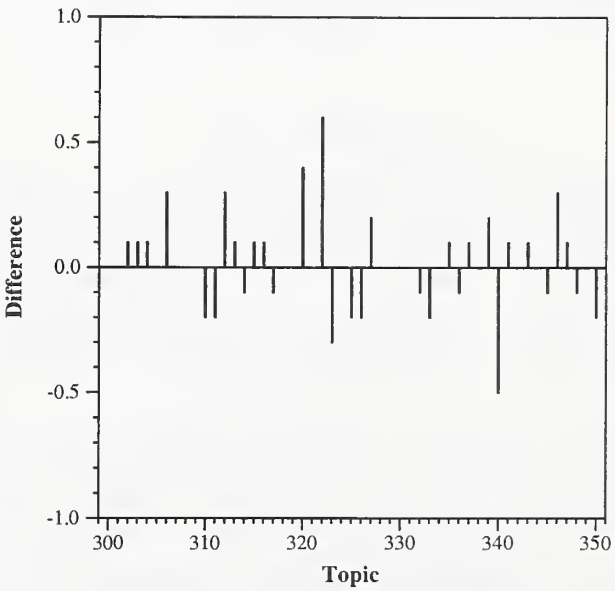
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	otc1
Number of Topics	50
Total number of documents over all topics	
Retrieved:	472
Relevant:	4611
Rel-ret:	275

Means over 50 topics	
Precision at 10 Docs:	0.5500
Relative-Precision at 10 Docs:	0.5700
Unranked-Avg-Precision at 10 Docs:	0.0973



Precision(10) Difference from Median per Topic

Evaluation Measures

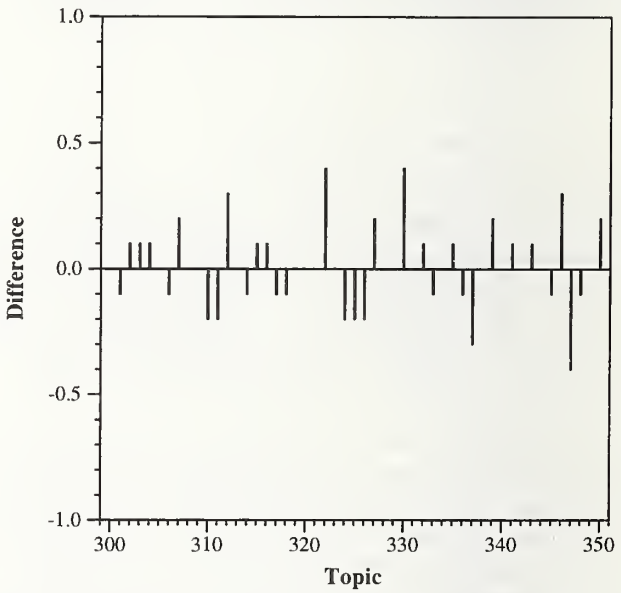
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	otc2
Number of Topics	50
Total number of documents over all topics	
Retrieved:	477
Relevant:	4611
Rel-ret:	272

Means over 50 topics	
Precision at 10 Docs:	0.5440
Relative-Precision at 10 Docs:	0.5587
Unranked-Avg-Precision at 10 Docs:	0.0916



Precision(10) Difference from Median per Topic

Evaluation Measures

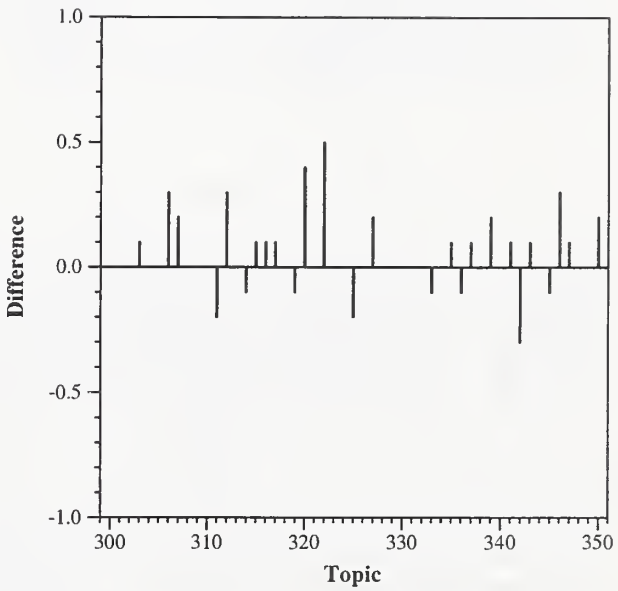
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	otc3
Number of Topics	50
Total number of documents over all topics	
Retrieved:	491
Relevant:	4611
Rel-ret:	290

Means over 50 topics	
Precision at 10 Docs:	0.5800
Relative-Precision at 10 Docs:	0.6020
Unranked-Avg-Precision at 10 Docs:	0.1067



Precision(10) Difference from Median per Topic

Evaluation Measures

Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	pirc7Ha
Number of Topics	50
Total number of documents over all topics	
Retrieved:	500
Relevant:	4611
Rel-ret:	213

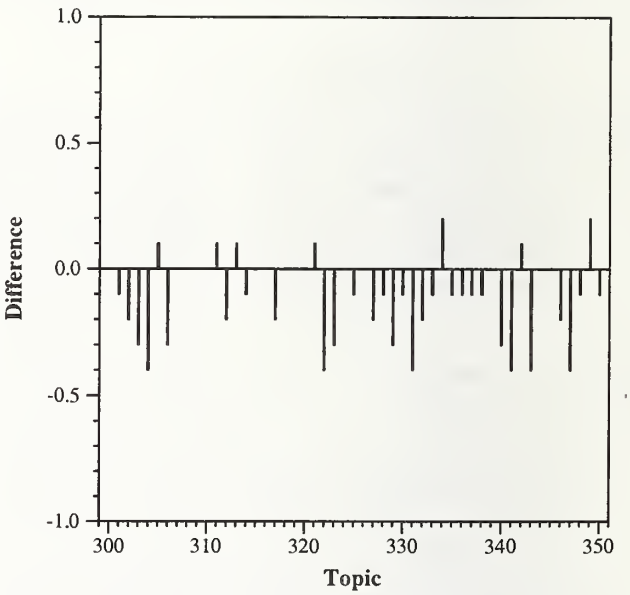
Means over 50 topics	
Precision at 10 Docs:	0.4260
Relative-Precision at 10 Docs:	0.4384
Unranked-Avg-Precision at 10 Docs:	0.0574

Evaluation Measures

Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.



Precision(10) Difference from Median per Topic

Summary Statistics	
Run Number	pirc7Hd
Number of Topics	50
Total number of documents over all topics	
Retrieved:	500
Relevant:	4611
Rel-ret:	168

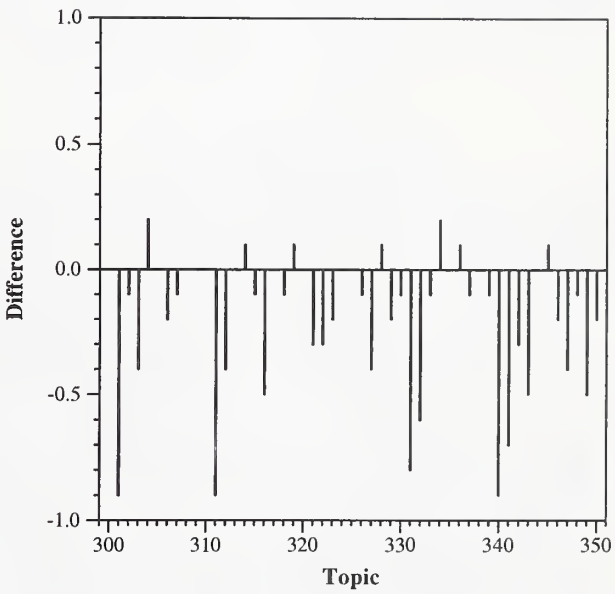
Means over 50 topics	
Precision at 10 Docs:	0.3360
Relative-Precision at 10 Docs:	0.3509
Unranked-Avg-Precision at 10 Docs:	0.0561

Evaluation Measures

Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

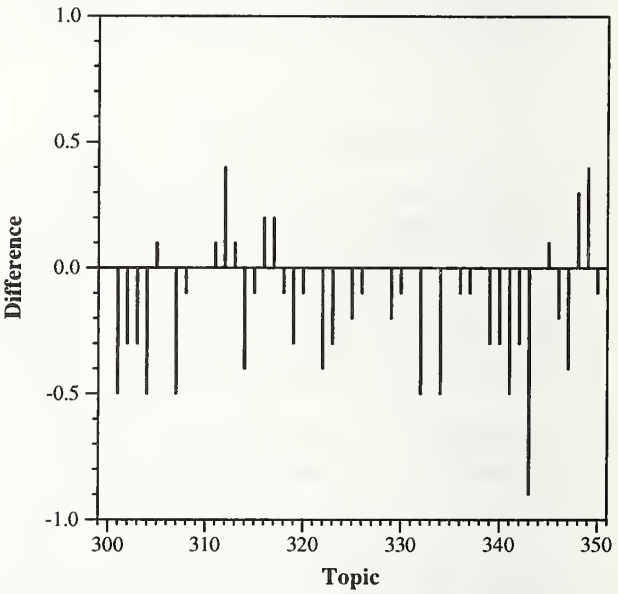
Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.



Precision(10) Difference from Median per Topic

Summary Statistics	
Run Number	pirc7Ht
Number of Topics	50
Total number of documents over all topics	
Retrieved:	500
Relevant:	4611
Rel-ret:	199

Means over 50 topics	
Precision at 10 Docs:	0.3980
Relative-Precision at 10 Docs:	0.4163
Unranked-Avg-Precision at 10 Docs:	0.0766



Precision(10) Difference from Median per Topic

Evaluation Measures

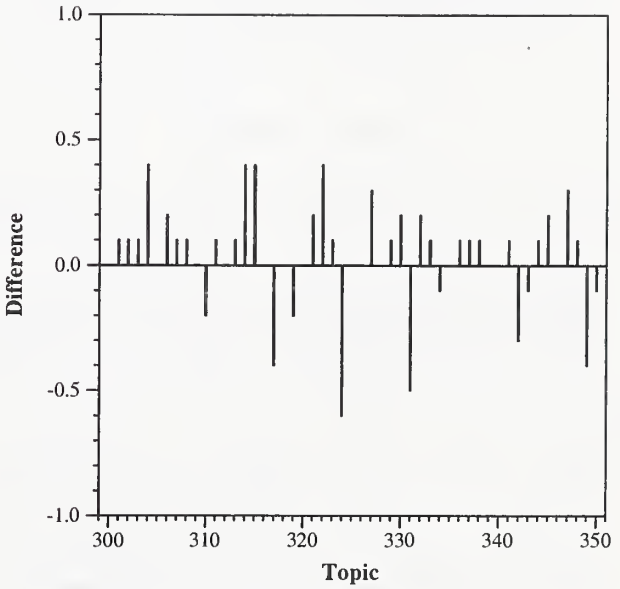
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	uwmt6h0
Number of Topics	50
Total number of documents over all topics	
Retrieved:	438
Relevant:	4611
Rel-ret:	286

Means over 50 topics	
Precision at 10 Docs:	0.5720
Relative-Precision at 10 Docs:	0.5977
Unranked-Avg-Precision at 10 Docs:	0.0902



Precision(10) Difference from Median per Topic

Evaluation Measures

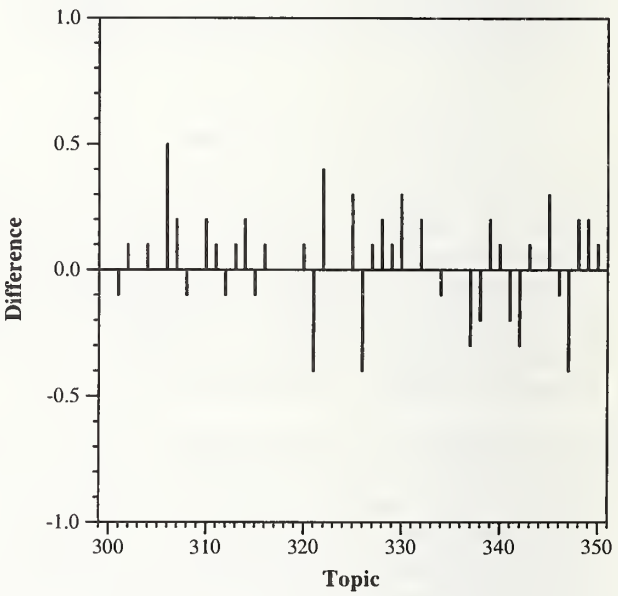
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	uwmt6h1
Number of Topics	50
Total number of documents over all topics	
Retrieved:	414
Relevant:	4611
Rel-ret:	284

Means over 50 topics	
Precision at 10 Docs:	0.5680
Relative-Precision at 10 Docs:	0.5834
Unranked-Avg-Precision at 10 Docs:	0.0982



Precision(10) Difference from Median per Topic

Evaluation Measures

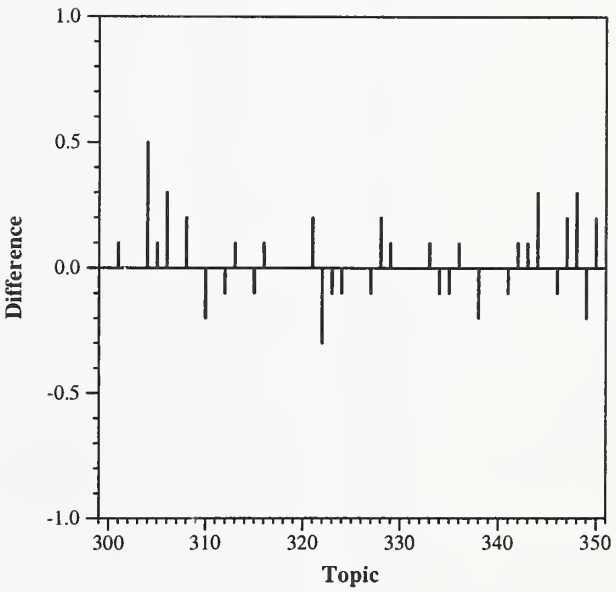
Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Summary Statistics	
Run Number	uwmt6h2
Number of Topics	50
Total number of documents over all topics	
Retrieved:	465
Relevant:	4611
Rel-ret:	282

Means over 50 topics	
Precision at 10 Docs:	0.5640
Relative-Precision at 10 Docs:	0.5951
Unranked-Avg-Precision at 10 Docs:	0.0997



Precision(10) Difference from Median per Topic

Evaluation Measures

Precision at 10: The percentage of documents retrieved in the top ten that are relevant. If fewer than 10 documents are retrieved, then all missing documents are assumed to be non-relevant. Precision considers each retrieved relevant document to be equally important, no matter if is retrieved for a query with 500 relevant documents or a query with two relevant documents.

Relative Precision at 10: The precision after ten documents relative to the maximum precision possible at that point. For example, if there are 4 relevant documents and 3 of those are retrieved, then Precision at 10 is .3, but the Relative-Precision is .3/.4 or .75. Relative-Precision considers each query ro be equally important. Thus a query with only two relevant documents has the same maximum score of 1.0 as a query with ten or more relevant documents.

Unranked-Average Precision at 10: Similar to the standard TREC "average precision" measure, with all retrieved relevant documents getting a precision value of the retrieved set (i.e., $r/10$ where r is the number of relevant documents retrieved). All non-retrieved relevant documents get a precision value of 0. This measure is actually directed at unranked evaluation where the size of the retrieved set is under the control of the user, and is not completely appropriate for the High Precision track. It is included to gain more operational experience with the measure, and to see if it offers insights into the results.

Interactive Track: Aspects identified by the NIST assessors

Topic	Aspect#	Aspect gloss
303i	1	has inspired new cosmological theories
303i	2	study of gravitational lenses
303i	3	more precise estimate of scale, size, and age of universe
303i	4	picture of more distant galaxies/objects
303i	5	generally good, better, better than expected results
303i	6	contradicted existing cosmological theories
303i	7	supported existing cosmological theories
307i	1	China - Three Gorges, Yangtse, Sanxia
307i	2	Slovakia- Bos-Nagymaros/Gabcikova/Cunovo
307i	3	Kenya - Ewaso Ngiro
307i	4	Mexico - Rio Usumacinta
307i	5	Canada - James Bay/Great Whale
307i	6	Iran - Karun
307i	7	India - Narmada
307i	8	Kyrgyzstan - Naryn
307i	9	Chile - Panque/Bo-Bo/Bio-Bio
307i	10	Bulgaria - Chiara
307i	11	Argentina/Paraguay - Yacireta/Parana
307i	12	Columbia - Cordoba
307i	13	Vietnam - North
307i	14	Malaysia - Pergali
307i	15	Malaysia - Kelanta
307i	16	Turkey - Birecik
307i	17	Malaysia - Sarawak/Bakun
307i	18	Nepal - Arun
307i	19	Portugal - Vila Nova de Foz Coa
307i	20	China - Xiaolangdi
307i	21	Paraguay - Corpus Cristi
307i	22	Malaysia - Sabah
307i	23	Mekong
322i	1	stolen paintings with a ransom demand
322i	2	forged rare book published
322i	3	marketing of forged cameo art
322i	4	art works smuggled for sale abroad
322i	5	stolen art work offered for sale
322i	6	sale of forged art work
322i	7	theft and forgery in sale of art pieces
322i	8	imitation jewelry sold as art work

322i 9 plundered bronze head sold at auction
326i 1 Zairean ferry accident
326i 2 Neptune ferry sinks
326i 3 Korean ferry sinks west coast of South Korea
326i 4 Moby Prince ferry fire - (2.5 miles at sea)
326i 5 Herald Free Enterprise ferry off Belgian coast
326i 6 Ferry capsizes Port of Mombasa
326i 7 Estonian sinks
326i 8 Bangladesh ferrys sink in Bay of Bengal (Oct 94)
326i 9 Philippine ferry sinking (apparently in the Philippines)
339i 1 Alcav
339i 2 pivacetam
339i 3 oxiracetam
339i 4 tacrine - Cognex
339i 5 physostigmine
339i 6 Aviva
339i 7 velnacrine - Mentane
339i 8 selegiline (Eldepryl)
339i 9 Zofran (ondansetron)
339i 10 denbuflline
347i 1 Finland - saima ringed seal
347i 2 Brazil - golden lion tamarin
347i 3 Japan - Atlantic bluefish tuna, elephants
347i 4 Int'l Commission for Conservation of Atlantic Tuna - Atl. blue tuna
347i 5 Kenya - elephants
347i 6 Columbia - Andean condor
347i 7 South Africa - quagga, white rhino
347i 8 Belize - jaguar, black howler monkey
347i 9 Zimbabwe - rhino, elephants
347i 10 UK - capercaillie, tern, polecat, birds
347i 11 Oman - Arabian oryx
347i 12 EC - harp and ring seals
347i 13 Spain - white-headed duck
347i 14 Greece - elephants
347i 15 Worldwide Fund for Nature - sea birds (long-tailed guillenot, shag, fulmar, little auk, Gr. North. Diver), elephants, panda, rhino, Bengal tiger, Barasingha deer.
347i 16 Paraguay - teyu guazu iguana, cayman, boa constrictor
347i 17 Poland - bison
347i 18 Indonesia - wild monkeys, chimps, Sumatra tiger
347i 19 Cites - elephants
347i 20 New Zealand - birds
347i 21 Peru - vicuna
347i 22 Canada - cod
347i 23 India - tigers
347i 24 China - rhino, tiger

347i 25 Romania - European mink
347i 26 Zambia - elephant, black rhino

Interactive Track: Per search measures - aspectual precision, aspectual recall, and time

Site	Search	Searcher	Topic	System	#docs	Prec	Rec.	Time	Aspect coverage vector Aspect# 1 -> n ''1''=covered, ''0''=not
BrklyINT	P1-3	P1	307i	CHESHIRE	17	0.765	0.565	1200	11110101101011001101000
BrklyINT	P1-2	P1	322i	CHESHIRE	8	0.375	0.222	1200	000011000
BrklyINT	P1-1	P1	326i	CHESHIRE	8	0.875	0.667	1200	001110111
BrklyINT	P1-5	P1	303i	ZP	3	0.667	1.000	630	1111111
BrklyINT	P1-6	P1	339i	ZP	4	1.000	0.800	1200	0111111110
BrklyINT	P1-4	P1	347i	ZP	4	1.000	0.154	690	0100001001000000000100000
BrklyINT	P2-5	P2	303i	CHESHIRE	2	1.000	1.000	787	1111111
BrklyINT	P2-6	P2	339i	CHESHIRE	4	1.000	0.900	1200	1111111101
BrklyINT	P2-4	P2	347i	CHESHIRE	9	1.000	0.308	1200	0100001011000001100010100
BrklyINT	P2-3	P2	307i	ZP	9	0.889	0.348	1015	10001100110000011100000
BrklyINT	P2-2	P2	322i	ZP	3	0.667	0.222	990	000101000
BrklyINT	P2-1	P2	326i	ZP	3	0.667	0.333	1125	100010100
BrklyINT	P3-3	P3	307i	CHESHIRE	9	0.667	0.261	1200	10110100000100000000010
BrklyINT	P3-2	P3	322i	CHESHIRE	5	0.200	0.111	1200	000001000
BrklyINT	P3-1	P3	326i	CHESHIRE	6	1.000	0.667	1200	010110111
BrklyINT	P3-5	P3	303i	ZP	5	0.600	1.000	1200	1111111
BrklyINT	P3-6	P3	339i	ZP	7	0.571	0.900	1200	1111111101
BrklyINT	P3-4	P3	347i	ZP	13	0.846	0.462	1200	011100100100001101111101000
BrklyINT	P4-5	P4	303i	CHESHIRE	5	0.800	1.000	1200	1111111
BrklyINT	P4-6	P4	339i	CHESHIRE	4	1.000	0.900	1200	1111111101
BrklyINT	P4-4	P4	347i	CHESHIRE	6	0.833	0.269	1200	110000100110100001000000000
BrklyINT	P4-3	P4	307i	ZP	6	1.000	0.261	1200	11000100100000010100000
BrklyINT	P4-2	P4	322i	ZP	1	1.000	0.111	1200	000100000
BrklyINT	P4-1	P4	326i	ZP	4	0.750	0.333	1200	001010100
IBM	307i-1-NQ-IBM	1	307i	NQ	10	0.900	0.348	914	1101010010010000000101O
IBM	322i-1-NQ-IBM	1	322i	NQ	3	0.333	0.111	1161	000001000
IBM	326i-1-NQ-IBM	1	326i	NQ	4	0.750	0.333	1179	000001011
IBM	303i-1-ZP-IBM	1	303i	ZP	3	0.333	0.571	888	1111000
IBM	339i-1-ZP-IBM	1	339i	ZP	5	0.600	0.700	1161	1111110001
IBM	347i-1-ZP-IBM	1	347i	ZP	11	0.636	0.231	1152	010010100100001000100000O0
IBM	303i-2-NQ-IBM	2	303i	NQ	1	0.000	0.000	1173	0000000
IBM	339i-2-NQ-IBM	2	339i	NQ	4	1.000	0.900	1148	1111111101
IBM	347i-2-NQ-IBM	2	347i	NQ	3	0.667	0.154	1106	000000100000001000100100O0
IBM	307i-2-ZP-IBM	2	307i	ZP	7	0.857	0.217	1079	00100100100000010100000
IBM	322i-2-ZP-IBM	2	322i	ZP	3	0.000	0.000	1141	000000000
IBM	326i-2-ZP-IBM	2	326i	ZP	2	1.000	0.222	1167	000010100
IBM	307i-3-NQ-IBM	3	307i	NQ	8	0.750	0.174	986	01010100000100000000000
IBM	322i-3-NQ-IBM	3	322i	NQ	3	0.333	0.111	1182	000001000
IBM	326i-3-NQ-IBM	3	326i	NQ	4	1.000	0.111	1169	000000100
IBM	303i-3-ZP-IBM	3	303i	ZP	7	0.429	1.000	1078	1111111
IBM	339i-3-ZP-IBM	3	339i	ZP	5	0.800	0.700	927	1111110001
IBM	347i-3-ZP-IBM	3	347i	ZP	4	0.500	0.077	1112	000000100100000000000000000
IBM	303i-4-NQ-IBM	4	303i	NQ	3	0.667	0.286	990	0001100
IBM	339i-4-NQ-IBM	4	339i	NQ	4	0.750	0.600	880	1011001101

IBM	347i-4-NQ-IBM	4	347i NQ	4	0.250	0.038	1183	000000000000000000000000100
IBM	307i-4-ZP-IBM	4	307i ZP	10	0.600	0.261	1140	10100100100000100100000
IBM	322i-4-ZP-IBM	4	322i ZP	8	0.875	0.222	1147	000110000
IBM	326i-4-ZP-IBM	4	326i ZP	5	1.000	0.333	1096	000110100
INQ4iai	01inqiai303	01	303i AI	4	0.500	0.714	806	0011111
INQ4iai	01inqiai339	01	339i AI	3	0.667	0.400	906	1011000001
INQ4iai	01inqiai347	01	347i AI	4	0.000	0.000	850	000000000000000000000000000
INQ4iai	01inqizp307	01	307i ZP	9	0.778	0.261	1200	011001000001000010100000
INQ4iai	01inqizp322	01	322i ZP	2	0.500	0.111	1200	000100000
INQ4iai	01inqizp326	01	326i ZP	6	0.500	0.222	1200	000000110
INQ4iai	02inqiai303	02	303i AI	3	0.667	0.714	800	0011111
INQ4iai	02inqiai339	02	339i AI	7	0.429	0.700	1022	1111110001
INQ4iai	02inqiai347	02	347i AI	8	0.500	0.154	1064	000010100100100000000000000
INQ4iai	02inqizp307	02	307i ZP	13	0.692	0.348	1027	11001100100000001100001
INQ4iai	02inqizp322	02	322i ZP	6	0.000	0.000	1110	000000000
INQ4iai	02inqizp326	02	326i ZP	4	0.750	0.222	882	000010100
INQ4iai	03inqiai307	03	307i AI	8	0.625	0.174	1200	010010000001000000010000
INQ4iai	03inqiai322	03	322i AI	4	0.500	0.222	1200	000110000
INQ4iai	03inqiai326	03	326i AI	3	1.000	0.333	1200	010010100
INQ4iai	03inqizp303	03	303i ZP	3	1.000	1.000	933	1111111
INQ4iai	03inqizp339	03	339i ZP	7	0.714	0.900	753	1111111101
INQ4iai	03inqizp347	03	347i ZP	13	0.692	0.269	1089	000010001100000100010010l0
INQ4iai	04inqiai307	04	307i AI	12	0.917	0.435	954	11110100100011000101000
INQ4iai	04inqiai322	04	322i AI	8	0.125	0.111	1068	000010000
INQ4iai	04inqiai326	04	326i AI	2	0.500	0.111	502	010000000
INQ4iai	04inqizp303	04	303i ZP	5	0.400	1.000	751	1111111
INQ4iai	04inqizp339	04	339i ZP	4	1.000	0.900	657	1111111101
INQ4iai	04inqizp347	04	347i ZP	9	0.778	0.269	775	011000000100001110000001000
INQ4iai	06inqiai303	06	303i AI	1	1.000	0.714	1200	0011111
INQ4iai	06inqiai339	06	339i AI	2	0.500	0.100	1200	0001000000
INQ4iai	06inqiai347	06	347i AI	1	1.000	0.038	1200	000000100000000000000000000
INQ4iai	06inqizp307	06	307i ZP	5	0.800	0.174	1200	000011100000000001000000
INQ4iai	06inqizp322	06	322i ZP	1	0.000	0.000	1089	000000000
INQ4iai	06inqizp326	06	326i ZP	2	1.000	0.222	1200	000010100
INQ4iai	07inqiai303	07	303i AI	3	0.667	1.000	588	1111111
INQ4iai	07inqiai339	07	339i AI	3	1.000	0.600	996	1011001101
INQ4iai	07inqiai347	07	347i AI	8	0.500	0.154	1200	000010001100010000000000000
INQ4iai	07inqizp307	07	307i ZP	19	0.684	0.522	1125	11101111100001011100000
INQ4iai	07inqizp322	07	322i ZP	7	0.429	0.222	890	000110000
INQ4iai	07inqizp326	07	326i ZP	5	0.800	0.556	1160	000110111
INQ4iai	08inqiai307	08	307i AI	9	1.000	0.261	1200	010001001001100000001000
INQ4iai	08inqiai322	08	322i AI	1	0.000	0.000	891	000000000
INQ4iai	08inqiai326	08	326i AI	5	1.000	0.667	1200	110010111
INQ4iai	08inqizp303	08	303i ZP	4	0.750	1.000	789	1111111
INQ4iai	08inqizp339	08	339i ZP	5	0.800	0.600	525	1011001101
INQ4iai	08inqizp347	08	347i ZP	13	0.769	0.423	1200	010000101110001101100010001
INQ4iai	09inqiai307	09	307i AI	16	0.688	0.478	1200	11110101110100001001000
INQ4iai	09inqiai322	09	322i AI	5	0.400	0.111	1200	000010000
INQ4iai	09inqiai326	09	326i AI	5	1.000	0.556	1200	010010111
INQ4iai	09inqizp303	09	303i ZP	3	0.333	0.714	488	0011111
INQ4iai	09inqizp339	09	339i ZP	10	0.400	0.700	677	1111110001
INQ4iai	09inqizp347	09	347i ZP	7	0.857	0.192	435	000010100101000000000001000
INQ4iaip	10inqiaip303	10	303i AIP	3	0.667	1.000	562	1111111
INQ4iaip	10inqiaip339	10	339i AIP	4	1.000	0.700	618	1111110001
INQ4iaip	10inqiaip347	10	347i AIP	2	0.500	0.038	596	000000100000000000000000000
INQ4iaip	10inqizp307	10	307i ZP	9	0.667	0.174	900	10000101000000000000010
INQ4iaip	10inqizp322	10	322i ZP	0	0.000	0.000	1200	0

```
INQ4iaip 10inqizp326   10      326i ZP      12 1.000 0.444 1200 000110101
INQ4iaip 11inqiaip303  11      303i AIP      3 0.667 1.000  777 1111111
INQ4iaip 11inqiaip339  11      339i AIP      3 1.000 0.900  922 1111111101
INQ4iaip 11inqiaip347  11      347i AIP      6 0.833 0.192  924 10000010010011000000000000
INQ4iaip 11inqizp307   11      307i ZP       7 0.857 0.217 1110 10100100100000000100000
INQ4iaip 11inqizp322   11      322i ZP       1 1.000 0.111  726 000010000
INQ4iaip 11inqizp326   11      326i ZP       5 1.000 0.667  879 011110101
INQ4iaip 12inqiaip303  12      303i AIP      4 0.500 1.000 1200 1111111
INQ4iaip 12inqiaip339  12      339i AIP      5 0.800 0.700 1200 0111111100
INQ4iaip 12inqiaip347  12      347i AIP      6 0.833 0.154 1200 10001000010001000000000000
INQ4iaip 12inqizp307   12      307i ZP       3 0.667 0.087 1200 00000100000000010000000
INQ4iaip 12inqizp322   12      322i ZP       3 0.333 0.111 1200 000010000
INQ4iaip 12inqizp326   12      326i ZP       2 1.000 0.222 1200 000010100
INQ4iaip 13inqiaip307  13      307i AIP     10 0.900 0.348  790 01110101100100000001000
INQ4iaip 13inqiaip322  13      322i AIP      1 0.000 0.000  738 000000000
INQ4iaip 13inqiaip326  13      326i AIP      7 1.000 0.778 1200 110110111
INQ4iaip 13inqizp303   13      303i ZP       2 0.500 0.714  630 0011111
INQ4iaip 13inqizp339   13      339i ZP       6 0.667 0.700  810 0111111100
INQ4iaip 13inqizp347   13      347i ZP       7 1.000 0.269 1100 00000010010000110010001100
INQ4iaip 14inqiaip303  14      303i AIP      1 1.000 0.714  777 0011111
INQ4iaip 14inqiaip339  14      339i AIP      4 1.000 0.900  922 1111111101
INQ4iaip 14inqiaip347  14      347i AIP     10 0.400 0.192  924 00000010010000100010010000
INQ4iaip 14inqizp307   14      307i ZP       4 0.500 0.087 1110 00001100000000000000000
INQ4iaip 14inqizp322   14      322i ZP       1 1.000 0.111  726 000100000
INQ4iaip 14inqizp326   14      326i ZP       5 1.000 0.444  879 000110101
INQ4iaip 15inqiaip307  15      307i AIP      8 0.875 0.304  609 11101100100001000000000
INQ4iaip 15inqiaip322  15      322i AIP      4 0.250 0.111 1006 000100000
INQ4iaip 15inqiaip326  15      326i AIP      4 1.000 0.556  559 010010111
INQ4iaip 15inqizp303   15      303i ZP       3 1.000 1.000  435 1111111
INQ4iaip 15inqizp339   15      339i ZP       6 0.667 0.900  540 1111111101
INQ4iaip 15inqizp347   15      347i ZP       5 0.600 0.115  402 00001000010000010000000000
INQ4iaip 17inqiaip307  17      307i AIP     17 0.824 0.522 1200 11101101100101001001010
INQ4iaip 17inqiaip322  17      322i AIP      2 0.500 0.111 1200 000001000
INQ4iaip 17inqiaip326  17      326i AIP      7 1.000 0.667 1179 110010111
INQ4iaip 17inqizp303   17      303i ZP       2 1.000 1.000  520 1111111
INQ4iaip 17inqizp339   17      339i ZP       6 1.000 0.900  788 1111111101
INQ4iaip 17inqizp347   17      347i ZP      11 0.727 0.346 1200 00001010110000110010101000
INQ4iaip 19inqiaip307  19      307i AIP     11 0.909 0.435 1143 11100101100101000100010
INQ4iaip 19inqiaip322  19      322i AIP      2 0.500 0.111 1200 000001000
INQ4iaip 19inqiaip326  19      326i AIP      5 1.000 0.556  497 010010111
INQ4iaip 19inqizp303   19      303i ZP       2 1.000 1.000  544 1111111
INQ4iaip 19inqizp339   19      339i ZP       4 0.500 0.700  938 1111110001
INQ4iaip 19inqizp347   19      347i ZP       7 0.714 0.192  771 00000010010000010000100001
INQ4int  16inqiai307   16      307i AI       8 0.625 0.217 1121 0000010101010000000 1000
INQ4int  16inqiai322   16      322i AI       3 0.000 0.000 1200 000000000
INQ4int  16inqiai326   16      326i AI       5 1.000 0.556 1200 101010110
INQ4int  16inqiaip303  16      303i AIP      4 0.500 0.714 1177 0011111
INQ4int  16inqiaip339  16      339i AIP      7 0.571 0.900 1166 1111111101
INQ4int  16inqiaip347  16      347i AIP      6 0.833 0.192 1200 00001010110001000000000000
INQ4int  18inqiai307   18      307i AI      17 0.882 0.522 1200 11111101100101000001010
INQ4int  18inqiai322   18      322i AI      12 0.500 0.222 1200 000110000
INQ4int  18inqiai326   18      326i AI       6 0.833 0.667 1200 011010111
INQ4int  18inqiaip303  18      303i AIP      2 1.000 1.000 1200 1111111
INQ4int  18inqiaip339  18      339i AIP     13 0.538 1.000 1200 1111111111
INQ4int  18inqiaip347  18      347i AIP     11 0.636 0.308 1200 01000010011110000100000100
INQ4int  20inqiai303   20      303i AI       3 0.667 1.000 1200 1111111
INQ4int  20inqiai339   20      339i AI       4 0.750 0.900 1200 1111111101
```

INQ4int	20inqiai347	20	347i	AI	9	0.444	0.154	1200	000010101000000000000100000
INQ4int	20inqiaip307	20	307i	AIP	13	0.692	0.391	1200	10100101000101000101010
INQ4int	20inqiaip322	20	322i	AIP	3	0.333	0.111	1200	000010000
INQ4int	20inqiaip326	20	326i	AIP	5	0.800	0.444	1200	000010111
INQ4int	21inqiai303	21	303i	AI	1	1.000	0.714	1200	0011111
INQ4int	21inqiai339	21	339i	AI	4	0.500	0.400	1200	1011000001
INQ4int	21inqiai347	21	347i	AI	9	0.778	0.269	1143	000010101100010000000001100
INQ4int	21inqiaip307	21	307i	AIP	12	0.833	0.304	1200	11110101000001000000000
INQ4int	21inqiaip322	21	322i	AIP	3	0.333	0.111	1200	000001000
INQ4int	21inqiaip326	21	326i	AIP	3	1.000	0.333	1200	010010100
NMSU	T5P1	P1	303i	IV	5	0.400	1.000	604	1111111
NMSU	T6P1	P1	339i	IV	5	0.800	0.700	927	1011001111
NMSU	T4P1	P1	347i	IV	6	1.000	0.308	240	0100001001100010111000000000
NMSU	T3P1	P1	307i	ZP	7	0.857	0.217	587	10100100100000000100000
NMSU	T2P1	P1	322i	ZP	2	0.500	0.111	918	000100000
NMSU	T1P1	P1	326i	ZP	4	0.750	0.444	1193	000110101
NMSU	T3P2	P2	307i	IV	11	0.727	0.304	812	11110000100001000100000
NMSU	T2P2	P2	322i	IV	2	1.000	0.222	864	100010000
NMSU	T1P2	P2	326i	IV	6	1.000	0.667	790	011010111
NMSU	T5P2	P2	303i	ZP	3	1.000	1.000	696	1111111
NMSU	T6P2	P2	339i	ZP	5	0.800	0.900	606	1111111101
NMSU	T4P2	P2	347i	ZP	10	0.900	0.346	1135	0100101001000010001010100 1
NMSU	T3P3	P3	307i	IV	3	0.667	0.087	653	0100010000000000000000
NMSU	T2P3	P3	322i	IV	2	1.000	0.111	1015	000010000
NMSU	T1P3	P3	326i	IV	4	0.750	0.333	1001	001000110
NMSU	T5P3	P3	303i	ZP	2	1.000	1.000	549	1111111
NMSU	T6P3	P3	339i	ZP	3	1.000	0.600	833	1011001101
NMSU	T4P3	P3	347i	ZP	5	1.000	0.192	1200	000010010000000000000001011
NMSU	T5P4	P4	303i	IV	2	1.000	1.000	1091	1111111
NMSU	T6P4	P4	339i	IV	6	0.500	0.700	1053	0111111100
NMSU	T4P4	P4	347i	IV	5	1.000	0.231	962	000000100100101010100000000
NMSU	T3P4	P4	307i	ZP	6	0.667	0.174	1200	100100001010000000000000
NMSU	T2P4	P4	322i	ZP	2	0.500	0.111	1200	000010000
NMSU	T1P4	P4	326i	ZP	5	1.000	0.333	706	000010110
OHSU	KS307i	KS	307i	MG	13	0.692	0.391	1200	1101001000100100110100 0
OHSU	KS322i	KS	322i	MG	1	1.000	0.111	1140	000001000
OHSU	KS326i	KS	326i	MG	5	1.000	0.333	1168	010010100
OHSU	KS303i	KS	303i	ZP	2	1.000	1.000	945	1111111
OHSU	KS339i	KS	339i	ZP	6	0.667	0.700	785	1111110001
OHSU	KS347i	KS	347i	ZP	8	0.625	0.192	1203	000010100100000000100000 0
OHSU	LD307i	LD	307i	MG	7	0.714	0.217	952	00110000001000000101000
OHSU	LD322i	LD	322i	MG	1	1.000	0.111	1134	000001000
OHSU	LD326i	LD	326i	MG	3	1.000	0.444	1164	010010110
OHSU	LD303i	LD	303i	ZP	3	0.667	1.000	1095	1111111
OHSU	LD339i	LD	339i	ZP	4	1.000	0.900	1148	1111111101
OHSU	LD347i	LD	347i	ZP	6	0.833	0.192	1084	000010101000000100000100000
OHSU	LS303i	LS	303i	MG	4	0.750	1.000	1178	1111111
OHSU	LS339i	LS	339i	MG	5	0.600	0.500	1203	1011000011
OHSU	LS347i	LS	347i	MG	3	1.000	0.115	1200	000000010110000000000000000
OHSU	LS307i	LS	307i	ZP	9	1.000	0.348	1196	000101011011000011000000
OHSU	LS322i	LS	322i	ZP	3	0.667	0.222	1172	000110000
OHSU	LS326i	LS	326i	ZP	6	1.000	0.444	1181	100110100
OHSU	SM303i	SM	303i	MG	2	1.000	1.000	1191	1111111
OHSU	SM339i	SM	339i	MG	1	1.000	0.100	1124	0001000000
OHSU	SM347i	SM	347i	MG	4	1.000	0.154	1179	1000001000000000000011000000
OHSU	SM307i	SM	307i	ZP	5	1.000	0.217	1183	10100100100000000100000
OHSU	SM322i	SM	322i	ZP	4	0.750	0.333	1193	001010010

OHSU	SM326i	SM	326i	ZP	3	0.667	0.333	1131	100010100
city	p13-307	cmw	307i	ok	12	0.833	0.391	1218	11110100100000001101000
city	p13-322	cmw	322i	ok	5	0.200	0.111	1233	000001000
city	p13-326	cmw	326i	ok	9	0.889	0.556	1262	010010111
city	p13-303	cmw	303i	zp	2	1.000	1.000	1202	1111111
city	p13-339	cmw	339i	zp	6	0.667	0.300	1247	0001001100
city	p13-347	cmw	347i	zp	16	0.812	0.500	1244	0100001011100C0110111101001
city	p24-303	fam	303i	ok	8	0.250	0.714	1217	0011111
city	p24-339	fam	339i	ok	6	0.667	0.700	1181	1111110001
city	p24-347	fam	347i	ok	14	0.500	0.385	1103	0100001000100010111000011
city	p24-307	fam	307i	zp	10	0.800	0.304	1246	10100110110000000100000
city	p24-322	fam	322i	zp	0	0.000	0.000	1193	0
city	p24-326	fam	326i	zp	25	0.920	0.556	1299	000110111
city	p14-303	fmp	303i	ok	3	0.333	0.143	1189	0000100
city	p14-339	fmp	339i	ok	2	1.000	0.400	1252	1011000001
city	p14-347	fmp	347i	ok	5	0.400	0.154	1205	000001100000001000100000000
city	p14-307	fmp	307i	zp	3	0.667	0.087	1260	000001001000000000000000
city	p14-322	fmp	322i	zp	0	0.000	0.000	1239	0
city	p14-326	fmp	326i	zp	2	1.000	0.333	1181	000010110
city	p12-303	lmc	303i	ok	2	1.000	1.000	1200	1111111
city	p12-339	lmc	339i	ok	3	1.000	0.900	1160	1111111101
city	p12-347	lmc	347i	ok	3	1.000	0.115	1207	000000100100000000010000000
city	p12-307	lmc	307i	zp	2	1.000	0.087	1190	000001000000000000100000
city	p12-322	lmc	322i	zp	1	1.000	0.111	1252	000010000
city	p12-326	lmc	326i	zp	1	1.000	0.222	1190	000010100
city	p22-303	ooo	303i	ok	4	0.750	1.000	993	1111111
city	p22-339	ooo	339i	ok	2	1.000	0.400	946	1011000001
city	p22-347	ooo	347i	ok	6	0.833	0.231	995	00100010010000100010000100
city	p22-307	ooo	307i	zp	5	0.800	0.174	1034	10100100000000000100000
city	p22-322	ooo	322i	zp	1	1.000	0.111	1254	000010000
city	p22-326	ooo	326i	zp	1	1.000	0.222	1088	000010100
city	p11-307	rel	307i	ok	13	0.846	0.391	1278	11010110100001001001000
city	p11-322	rel	322i	ok	3	0.667	0.111	1125	000010000
city	p11-326	rel	326i	ok	11	0.909	0.444	1220	000010111
city	p11-303	rel	303i	zp	2	1.000	1.000	1256	1111111
city	p11-339	rel	339i	zp	6	0.833	0.700	1221	1011001111
city	p11-347	rel	347i	zp	4	0.750	0.192	1176	010000000110000000110000000
city	p21-307	tak	307i	ok	4	1.000	0.174	1063	10100100100000000000000
city	p21-322	tak	322i	ok	1	0.000	0.000	1175	000000000
city	p21-326	tak	326i	ok	6	0.833	0.444	885	010000111
city	p21-303	tak	303i	zp	1	1.000	0.714	933	0011111
city	p21-339	tak	339i	zp	4	1.000	0.600	1128	1011001101
city	p21-347	tak	347i	zp	2	1.000	0.077	947	000000000100100000000000000
city	p23-307	wag	307i	ok	17	0.824	0.391	1193	11101100100001000101000
city	p23-322	wag	322i	ok	5	0.200	0.111	1236	000100000
city	p23-326	wag	326i	ok	5	1.000	0.333	1268	000010101
city	p23-303	wag	303i	zp	4	0.500	1.000	1253	1111111
city	p23-339	wag	339i	zp	6	0.833	0.700	1231	1011001111
city	p23-347	wag	347i	zp	6	0.833	0.154	1205	000010101000000000000000100
rmit	r1-s1-x-307	s1	307i	x	17	0.824	0.435	1200	11100101100001001101000
rmit	r1-s1-x-322	s1	322i	x	2	0.500	0.111	575	000100000
rmit	r1-s1-x-326	s1	326i	x	6	1.000	0.667	782	110001111
rmit	r1-s1-zp-303	s1	303i	zp	2	1.000	1.000	698	1111111
rmit	r1-s1-zp-339	s1	339i	zp	8	0.750	0.900	1046	1111111101
rmit	r1-s1-zp-347	s1	347i	zp	14	0.500	0.308	1200	011000000000000110010101010
rmit	r1-s2-x-303	s2	303i	x	3	1.000	1.000	681	1111111
rmit	r1-s2-x-339	s2	339i	x	4	0.750	0.700	1200	1111110001

rmit	r1-s2-x-347	s2	347i	x	6	0.833	0.231	1138	0100001000000011001010000
rmit	r1-s2-zp-307	s2	307i	zp	17	0.765	0.478	1200	11110110100000011100010
rmit	r1-s2-zp-322	s2	322i	zp	3	1.000	0.333	1064	001001010
rmit	r1-s2-zp-326	s2	326i	zp	5	1.000	0.667	1200	011011110
rmit	r1-s3-x-307	s3	307i	x	6	1.000	0.217	1118	1100010000100000000010
rmit	r1-s3-x-322	s3	322i	x	2	0.500	0.111	1200	100000000
rmit	r1-s3-x-326	s3	326i	x	5	1.000	0.556	1003	011000111
rmit	r1-s3-zp-303	s3	303i	zp	4	0.750	0.714	1061	0011111
rmit	r1-s3-zp-339	s3	339i	zp	4	0.500	0.700	984	0111111100
rmit	r1-s3-zp-347	s3	347i	zp	8	0.750	0.231	1126	0000100001000000000001101001
rmit	r1-s4-x-303	s4	303i	x	2	0.500	0.714	1103	0011111
rmit	r1-s4-x-339	s4	339i	x	4	0.750	0.700	1200	0111111100
rmit	r1-s4-x-347	s4	347i	x	2	1.000	0.154	1021	0100000000100100010000000
rmit	r1-s4-zp-307	s4	307i	zp	6	0.833	0.217	1033	01000101000000000010010
rmit	r1-s4-zp-322	s4	322i	zp	4	0.500	0.111	938	000001000
rmit	r1-s4-zp-326	s4	326i	zp	7	1.000	0.333	714	000010101
rutint1	s001.307i	s001	307i	ruinq1	9	0.889	0.304	1315	01000101100101000001000
rutint1	s001.322i	s001	322i	ruinq1	1	0.000	0.000	1095	000000000
rutint1	s001.326i	s001	326i	ruinq1	3	1.000	0.444	575	010010110
rutint1	s002.303i	s002	303i	ruinq1	2	1.000	1.000	970	1111111
rutint1	s002.339i	s002	339i	ruinq1	4	0.750	0.700	814	1111110001
rutint1	s002.347i	s002	347i	ruinq1	4	0.750	0.115	1264	0100000010000010000000000
rutint1	s005.307i	s005	307i	ruinq1	8	0.750	0.261	1012	00010100101001001000000
rutint1	s005.322i	s005	322i	ruinq1	1	0.000	0.000	1059	000000000
rutint1	s005.326i	s005	326i	ruinq1	5	1.000	0.556	1269	010010111
rutint1	s006.303i	s006	303i	ruinq1	4	0.500	1.000	1275	1111111
rutint1	s006.339i	s006	339i	ruinq1	9	0.667	0.900	1222	1111111101
rutint1	s006.347i	s006	347i	ruinq1	9	0.778	0.269	1285	0100001001000011001000010
rutint2	s003.307i	s003	307i	ruinq2	15	0.867	0.435	1207	0101011010100000110110
rutint2	s003.322i	s003	322i	ruinq2	7	0.429	0.222	1323	000010100
rutint2	s003.326i	s003	326i	ruinq2	7	0.857	0.778	1128	010111111
rutint2	s004.303i	s004	303i	ruinq2	6	0.500	1.000	899	1111111
rutint2	s004.339i	s004	339i	ruinq2	6	0.667	0.900	1213	1111111101
rutint2	s004.347i	s004	347i	ruinq2	4	1.000	0.231	1257	0000101001001010001000000
rutint2	s007.307i	s007	307i	ruinq2	12	0.667	0.348	1252	1101010010110000000 1000
rutint2	s007.322i	s007	322i	ruinq2	3	0.333	0.111	1088	000001000
rutint2	s007.326i	s007	326i	ruinq2	10	0.900	0.444	1269	011010100
rutint2	s008.303i	s008	303i	ruinq2	5	0.400	1.000	1272	1111111
rutint2	s008.339i	s008	339i	ruinq2	4	0.750	0.700	1203	1111110001
rutint2	s008.347i	s008	347i	ruinq2	8	0.500	0.154	1194	0000101000000001000000100
unc6ia	1_303	irisa1i	303i	ZP	2	1.000	1.000	1200	1111111
unc6ia	1_339	irisa1i	339i	ZP	3	1.000	0.700	1200	1111110001
unc6ia	1_347	irisa1i	347i	ZP	5	0.800	0.154	1200	0100000010000010000001000
unc6ia	1_307	irisa1i	307i	irisa	8	0.875	0.304	1200	01000101100101000100000
unc6ia	1_322	irisa1i	322i	irisa	0	0.000	0.000	1200	0
unc6ia	1_326	irisa1i	326i	irisa	5	1.000	0.667	1200	011011110
unc6ia	2_307	irisa2i	307i	ZP	12	0.750	0.391	1200	10100101100100010101000
unc6ia	2_322	irisa2i	322i	ZP	11	0.455	0.333	1200	000011001
unc6ia	2_326	irisa2i	326i	ZP	5	1.000	0.556	1200	000011111
unc6ia	2_303	irisa2i	303i	irisa	5	0.600	1.000	1200	1111111
unc6ia	2_339	irisa2i	339i	irisa	3	0.667	0.400	1200	1011000001
unc6ia	2_347	irisa2i	347i	irisa	4	0.750	0.154	1200	0000001001000010001000000
unc6ia	3_303	irisa3i	303i	ZP	4	0.500	1.000	1200	1111111
unc6ia	3_339	irisa3i	339i	ZP	7	0.857	0.700	1200	1011001111
unc6ia	3_347	irisa3i	347i	ZP	16	0.812	0.423	1200	0100101011000001011110 1000
unc6ia	3_307	irisa3i	307i	irisa	9	1.000	0.348	1200	10010100101000001100010
unc6ia	3_322	irisa3i	322i	irisa	0	0.000	0.000	1200	0

```
unc6ia   3_326        irisa3i 326i irisa     8 0.750 0.556 1200 110000111
unc6ia   4_307        irisa4i 307i ZP       14 0.714 0.435 1200 111␣1001001000001100001
unc6ia   4_322        irisa4i 322i ZP        5 0.600 0.222 1200 010010000
unc6ia   4_326        irisa4i 326i ZP        4 0.500 0.222 1200 000010100
unc6ia   4_303        irisa4i 303i irisa     6 0.500 1.000 1200 1111111
unc6ia   4_339        irisa4i 339i irisa     5 1.000 0.900 1200 1111111101
unc6ia   4_347        irisa4i 347i irisa     0 0.000 0.000 1200 0
unc6ip   5_307        irisp5i 307i ZP       12 0.833 0.435 1200 01010101101000011100001
unc6ip   5_322        irisp5i 322i ZP        3 1.000 0.111 1200 000010000
unc6ip   5_326        irisp5i 326i ZP        3 1.000 0.444 1200 100010110
unc6ip   5_303        irisp5i 303i irisp     2 1.000 1.000 1200 1111111
unc6ip   5_339        irisp5i 339i irisp     5 0.800 0.900 1200 1111111101
unc6ip   5_347        irisp5i 347i irisp     7 0.714 0.154 1200 0100000001000000000010000100
unc6ip   6_303        irisp6i 303i ZP        6 0.167 0.714 1200 0011111
unc6ip   6_339        irisp6i 339i ZP        3 1.000 0.600 1200 1011001101
unc6ip   6_347        irisp6i 347i ZP        8 0.625 0.231 1200 011000100000001000000100100
unc6ip   6_307        irisp6i 307i irisp     0 0.000 0.000 1200 0
unc6ip   6_322        irisp6i 322i irisp     2 1.000 0.111 1200 000010000
unc6ip   6_326        irisp6i 326i irisp     1 1.000 0.222 1200 000010100
unc6ip   7_307        irisp7i 307i ZP        5 0.800 0.174 1200 000001000100000010000010
unc6ip   7_322        irisp7i 322i ZP        9 0.667 0.111 1200 000010000
unc6ip   7_326        irisp7i 326i ZP        3 1.000 0.333 1200 010010100
unc6ip   7_303        irisp7i 303i irisp     3 1.000 1.000 1200 1111111
unc6ip   7_339        irisp7i 339i irisp     5 1.000 0.900 1200 1111111101
unc6ip   7_347        irisp7i 347i irisp    10 1.000 0.346 1200 11000010010010001010100100
unc6ip   8_303        irisp8i 303i ZP        9 0.444 1.000 1200 1111111
unc6ip   8_339        irisp8i 339i ZP        7 0.857 1.000 1200 1111111111
unc6ip   8_347        irisp8i 347i ZP        8 0.875 0.308 1200 01101000000000100010101100
unc6ip   8_307        irisp8i 307i irisp    22 0.909 0.522 1200 11011101101011001100000
unc6ip   8_322        irisp8i 322i irisp     1 0.000 0.000 1200 000000000
unc6ip   8_326        irisp8i 326i irisp     3 1.000 0.444 1200 000010111
```

Interactive Track: ANOVA for the cross-site model, output from SAS's PROC GLM

General Linear Models Procedure
Class Level Information

Class	Levels	Values
SITE	10	BrklyINT IBM INQ4iai INQ4iaip NMSU OHSU city rmit unc6ia unc6ip
TOPIC BLOCK	3	1 2 3

Number of observations in data set = 78

General Linear Models Procedure

Dependent Variable: Z Recall E-C

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	0.4395612	0.0399601	2.65	0.0071
Error	66	0.9944562	0.0150675		
Corrected Total	77	1.4340174			

R-Square	C.V.	Root MSE	DRECALL Mean
0.306524	-682.9163	0.1227	-0.0180

Source	DF	Type III SS	Mean Square	F Value	Pr > F
SITE	9	0.3490355	0.0387817	2.57	0.0133
TOPIC BLOCK	2	0.0905257	0.0452629	3.00	0.0564

Interactive Track: Tukey's Studentized Range (HSD) Test for the cross-site model

TREC-6 Interactive Experiment
Case 1: Assume errors are independent and sd constant
model drecall = site topblock

General Linear Models Procedure

Tukey's Studentized Range (HSD) Test for variable: DRECALL

NOTE: This test controls the type I experimentwise error rate,
but generally has a higher type II error rate than REGWQ.

Alpha= 0.05 df= 66 MSE= 0.015068
Critical Value of Studentized Range= 4.630
Minimum Significant Difference= 0.2139
WARNING: Cell sizes are not equal.
Harmonic Mean of cell sizes= 7.058824

Means with the same letter are not significantly different.

Tukey Grouping	Mean	N	SITE
A	0.07883	6	BrklyINT
A			
A	0.06158	12	INQ4iaip
A			
A	0.01958	6	NMSU
A			
A	0.01896	12	city
A			
A	0.01150	6	unc6ip
A			
A	-0.03300	6	rmit
A			
A	-0.06725	6	unc6ia
A			
A	-0.08667	12	INQ4iai
A			
A	-0.11400	6	IBM
A			
A	-0.11708	6	OHSU

Interactive Track: Tukey's Studentized Range (HSD)

TREC-6 Interactive Experiment
Case 1: Assume errors are independent and sd constant
model drecall = site topblock

General Linear Models Procedure

Tukey's Studentized Range (HSD) Test for variable: DRECALL

NOTE: This test controls the type I experimentwise error rate.

Alpha= 0.05 Confidence= 0.95 df= 66 MSE= 0.015068
Critical Value of Studentized Range= 4.630

Comparisons significant at the 0.05 level are indicated by '***'.

SITE Comparison	Simultaneous Lower Confidence Limit	Difference Between Means	Simultaneous Upper Confidence Limit
BrklyINT - INQ4iaip	-0.18371	0.01725	0.21821
BrklyINT - NMSU	-0.17279	0.05925	0.29129
BrklyINT - city	-0.14108	0.05987	0.26083
BrklyINT - unc6ip	-0.16471	0.06733	0.29938
BrklyINT - rmit	-0.12021	0.11183	0.34388
BrklyINT - unc6ia	-0.08596	0.14608	0.37813
BrklyINT - INQ4iai	-0.03546	0.16550	0.36646
BrklyINT - IBM	-0.03921	0.19283	0.42488
BrklyINT - OHSU	-0.03613	0.19592	0.42796
INQ4iaip - BrklyINT	-0.21821	-0.01725	0.18371
INQ4iaip - NMSU	-0.15896	0.04200	0.24296
INQ4iaip - city	-0.12145	0.04262	0.20670
INQ4iaip - unc6ip	-0.15087	0.05008	0.25104
INQ4iaip - rmit	-0.10637	0.09458	0.29554
INQ4iaip - unc6ia	-0.07212	0.12883	0.32979
INQ4iaip - INQ4iai	-0.01583	0.14825	0.31233
INQ4iaip - IBM	-0.02537	0.17558	0.37654
INQ4iaip - OHSU	-0.02229	0.17867	0.37962
NMSU - BrklyINT	-0.29129	-0.05925	0.17279
NMSU - INQ4iaip	-0.24296	-0.04200	0.15896
NMSU - city	-0.20033	0.00062	0.20158
NMSU - unc6ip	-0.22396	0.00808	0.24013

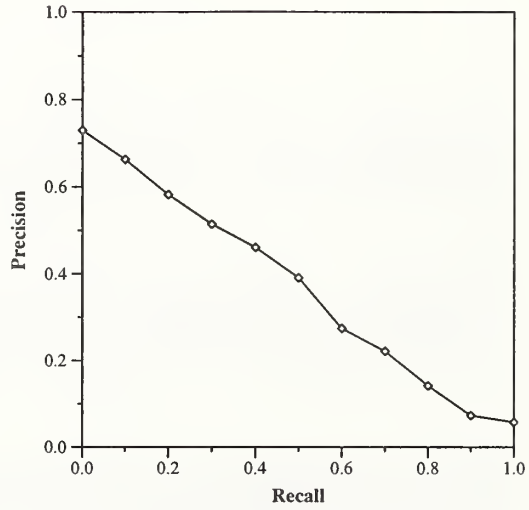
NMSU	- rmit	-0.17946	0.05258	0.28463
NMSU	- unc6ia	-0.14521	0.08683	0.31888
NMSU	- INQ4iai	-0.09471	0.10625	0.30721
NMSU	- IBM	-0.09846	0.13358	0.36563
NMSU	- OHSU	-0.09538	0.13667	0.36871
city	- BrklyINT	-0.26083	-0.05987	0.14108
city	- INQ4iaip	-0.20670	-0.04262	0.12145
city	- NMSU	-0.20158	-0.00062	0.20033
city	- unc6ip	-0.19350	0.00746	0.20841
city	- rmit	-0.14900	0.05196	0.25291
city	- unc6ia	-0.11475	0.08621	0.28716
city	- INQ4iai	-0.05845	0.10563	0.26970
city	- IBM	-0.06800	0.13296	0.33391
city	- OHSU	-0.06491	0.13604	0.33700
unc6ip	- BrklyINT	-0.29938	-0.06733	0.16471
unc6ip	- INQ4iaip	-0.25104	-0.05008	0.15087
unc6ip	- NMSU	-0.24013	-0.00808	0.22396
unc6ip	- city	-0.20841	-0.00746	0.19350
unc6ip	- rmit	-0.18754	0.04450	0.27654
unc6ip	- unc6ia	-0.15329	0.07875	0.31079
unc6ip	- INQ4iai	-0.10279	0.09817	0.29912
unc6ip	- IBM	-0.10654	0.12550	0.35754
unc6ip	- OHSU	-0.10346	0.12858	0.36063
rmit	- BrklyINT	-0.34388	-0.11183	0.12021
rmit	- INQ4iaip	-0.29554	-0.09458	0.10637
rmit	- NMSU	-0.28463	-0.05258	0.17946
rmit	- city	-0.25291	-0.05196	0.14900
rmit	- unc6ip	-0.27654	-0.04450	0.18754
rmit	- unc6ia	-0.19779	0.03425	0.26629
rmit	- INQ4iai	-0.14729	0.05367	0.25462
rmit	- IBM	-0.15104	0.08100	0.31304
rmit	- OHSU	-0.14796	0.08408	0.31613
unc6ia	- BrklyINT	-0.37813	-0.14608	0.08596
unc6ia	- INQ4iaip	-0.32979	-0.12883	0.07212
unc6ia	- NMSU	-0.31888	-0.08683	0.14521
unc6ia	- city	-0.28716	-0.08621	0.11475
unc6ia	- unc6ip	-0.31079	-0.07875	0.15329
unc6ia	- rmit	-0.26629	-0.03425	0.19779
unc6ia	- INQ4iai	-0.18154	0.01942	0.22037
unc6ia	- IBM	-0.18529	0.04675	0.27879
unc6ia	- OHSU	-0.18221	0.04983	0.28188

INQ4iai	- BrklyINT	-0.36646	-0.16550	0.03546
INQ4iai	- INQ4iaip	-0.31233	-0.14825	0.01583
INQ4iai	- NMSU	-0.30721	-0.10625	0.09471
INQ4iai	- city	-0.26970	-0.10563	0.05845
INQ4iai	- unc6ip	-0.29912	-0.09817	0.10279
INQ4iai	- rmit	-0.25462	-0.05367	0.14729
INQ4iai	- unc6ia	-0.22037	-0.01942	0.18154
INQ4iai	- IBM	-0.17362	0.02733	0.22829
INQ4iai	- OHSU	-0.17054	0.03042	0.23137
IBM	- BrklyINT	-0.42488	-0.19283	0.03921
IBM	- INQ4iaip	-0.37654	-0.17558	0.02537
IBM	- NMSU	-0.36563	-0.13358	0.09846
IBM	- city	-0.33391	-0.13296	0.06800
IBM	- unc6ip	-0.35754	-0.12550	0.10654
IBM	- rmit	-0.31304	-0.08100	0.15104
IBM	- unc6ia	-0.27879	-0.04675	0.18529
IBM	- INQ4iai	-0.22829	-0.02733	0.17362
IBM	- OHSU	-0.22896	0.00308	0.23513
OHSU	- BrklyINT	-0.42796	-0.19592	0.03613
OHSU	- INQ4iaip	-0.37962	-0.17867	0.02229
OHSU	- NMSU	-0.36871	-0.13667	0.09538
OHSU	- city	-0.33700	-0.13604	0.06491
OHSU	- unc6ip	-0.36063	-0.12858	0.10346
OHSU	- rmit	-0.31613	-0.08408	0.14796
OHSU	- unc6ia	-0.28188	-0.04983	0.18221
OHSU	- INQ4iai	-0.23137	-0.03042	0.17054
OHSU	- IBM	-0.23513	-0.00308	0.22896

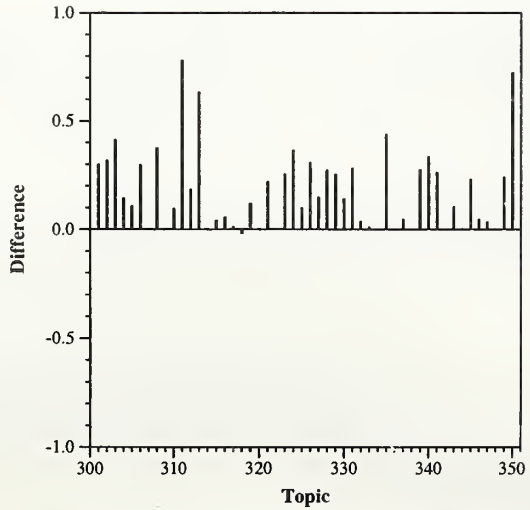
Summary Statistics	
Run Number	genlp1
Run Description	manual
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	1254

Recall Level Precision Averages	
Recall	Precision
0.00	0.7299
0.10	0.6632
0.20	0.5823
0.30	0.5139
0.40	0.4604
0.50	0.3910
0.60	0.2743
0.70	0.2216
0.80	0.1422
0.90	0.0734
1.00	0.0579
Average precision over all relevant docs	
non-interpolated	0.3555

Document Level Averages	
	Precision
At 5 docs	0.4553
At 10 docs	0.3809
At 15 docs	0.3447
At 20 docs	0.3170
At 30 docs	0.2716
At 100 docs	0.1615
At 200 docs	0.0999
At 500 docs	0.0477
At 1000 docs	0.0267
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.3332



Recall-Precision Curve

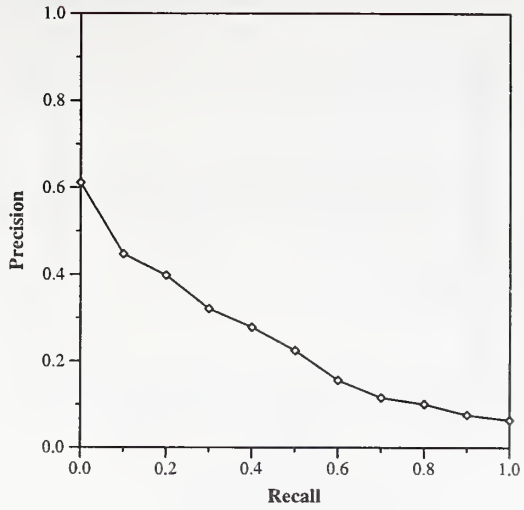


Difference from Median in Average Precision per Topic

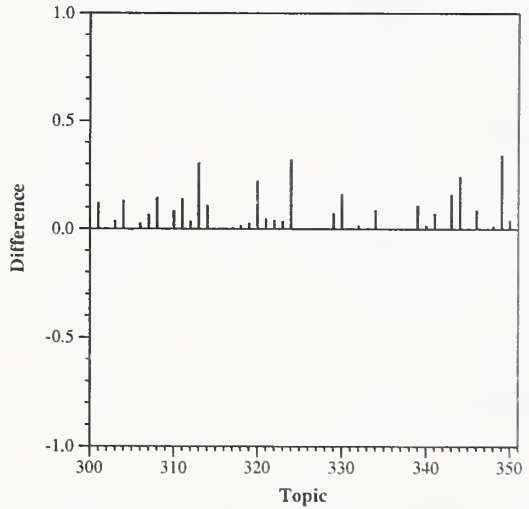
Summary Statistics	
Run Number	genlp2
Run Description	automatic
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	1056

Recall Level Precision Averages	
Recall	Precision
0.00	0.6114
0.10	0.4472
0.20	0.3982
0.30	0.3212
0.40	0.2784
0.50	0.2247
0.60	0.1561
0.70	0.1160
0.80	0.1010
0.90	0.0758
1.00	0.0639
Average precision over all relevant docs	
non-interpolated	0.2352

Document Level Averages	
	Precision
At 5 docs	0.3319
At 10 docs	0.2915
At 15 docs	0.2411
At 20 docs	0.2245
At 30 docs	0.1858
At 100 docs	0.1094
At 200 docs	0.0731
At 500 docs	0.0387
At 1000 docs	0.0225
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2588



Recall-Precision Curve

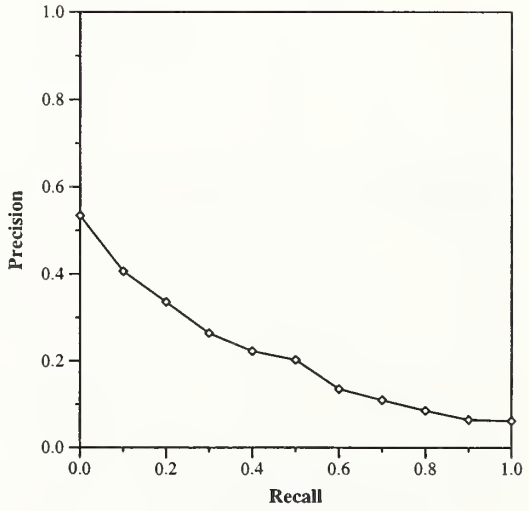


Difference from Median in Average Precision per Topic

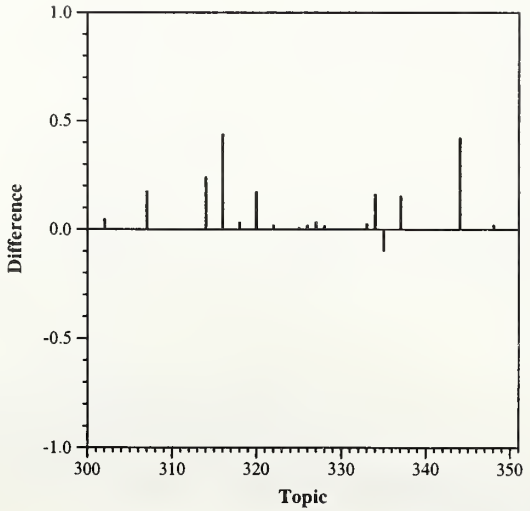
Summary Statistics	
Run Number	genlp3
Run Description	automatic
Number of Topics	47
Total number of documents over all topics	
Retrieved:	46014
Relevant:	1591
Rel-ret:	835

Recall Level Precision Averages	
Recall	Precision
0.00	0.5342
0.10	0.4067
0.20	0.3360
0.30	0.2641
0.40	0.2227
0.50	0.2026
0.60	0.1355
0.70	0.1100
0.80	0.0856
0.90	0.0647
1.00	0.0621
Average precision over all relevant docs	
non-interpolated	0.2051

Document Level Averages	
	Precision
At 5 docs	0.2681
At 10 docs	0.2383
At 15 docs	0.2156
At 20 docs	0.1904
At 30 docs	0.1567
At 100 docs	0.0885
At 200 docs	0.0581
At 500 docs	0.0291
At 1000 docs	0.0178
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2239



Recall-Precision Curve

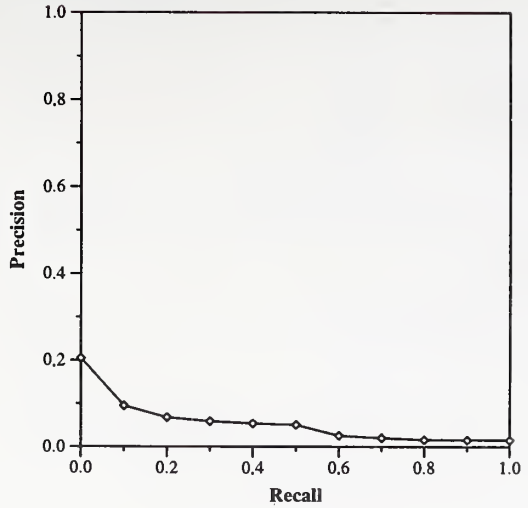


Difference from Median in Average Precision per Topic

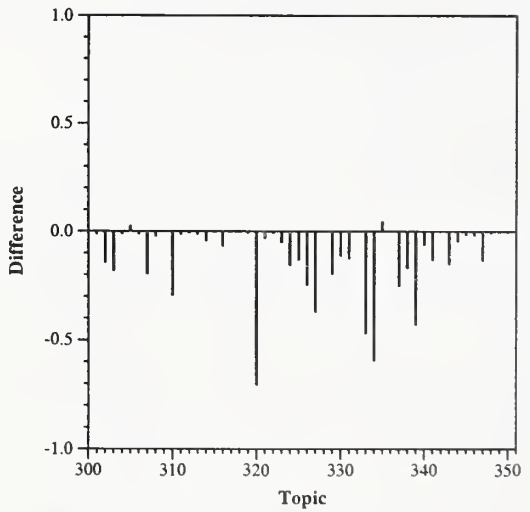
Summary Statistics	
Run Number	Gla6DS1
Run Description	automatic
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	515

Recall Level Precision Averages	
Recall	Precision
0.00	0.2049
0.10	0.0950
0.20	0.0678
0.30	0.0592
0.40	0.0539
0.50	0.0507
0.60	0.0257
0.70	0.0205
0.80	0.0163
0.90	0.0156
1.00	0.0156
Average precision over all relevant docs	
non-interpolated	0.0473

Document Level Averages	
	Precision
At 5 docs	0.0894
At 10 docs	0.0638
At 15 docs	0.0539
At 20 docs	0.0553
At 30 docs	0.0504
At 100 docs	0.0330
At 200 docs	0.0240
At 500 docs	0.0147
At 1000 docs	0.0110
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0530



Recall-Precision Curve

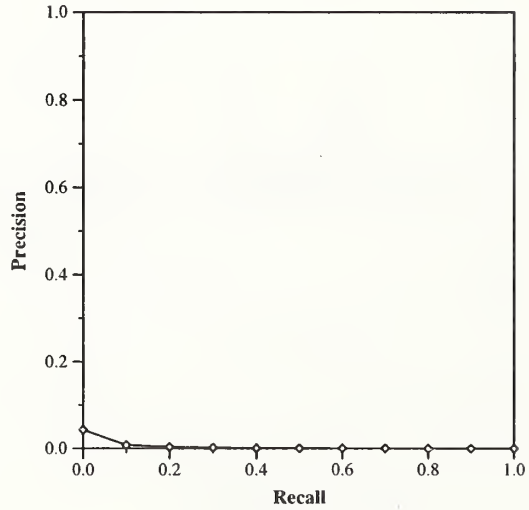


Difference from Median in Average Precision per Topic

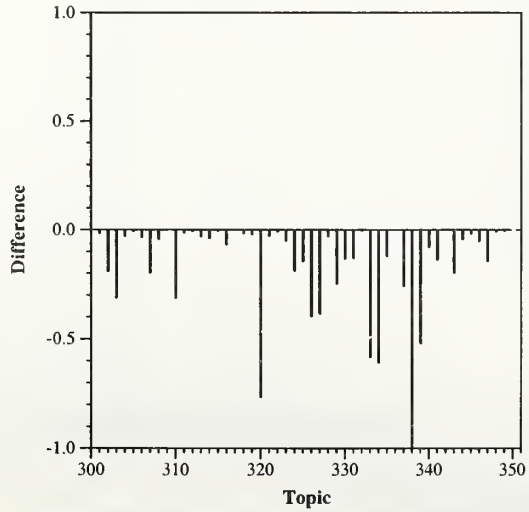
Summary Statistics	
Run Number	Gla6DS2
Run Description	automatic
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	148

Recall Level Precision Averages	
Recall	Precision
0.00	0.0429
0.10	0.0081
0.20	0.0036
0.30	0.0022
0.40	0.0008
0.50	0.0007
0.60	0.0003
0.70	0.0003
0.80	0.0000
0.90	0.0000
1.00	0.0000
Average precision over all relevant docs	
non-interpolated	0.0032

Document Level Averages	
	Precision
At 5 docs	0.0255
At 10 docs	0.0191
At 15 docs	0.0142
At 20 docs	0.0149
At 30 docs	0.0128
At 100 docs	0.0077
At 200 docs	0.0060
At 500 docs	0.0040
At 1000 docs	0.0031
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0075



Recall-Precision Curve

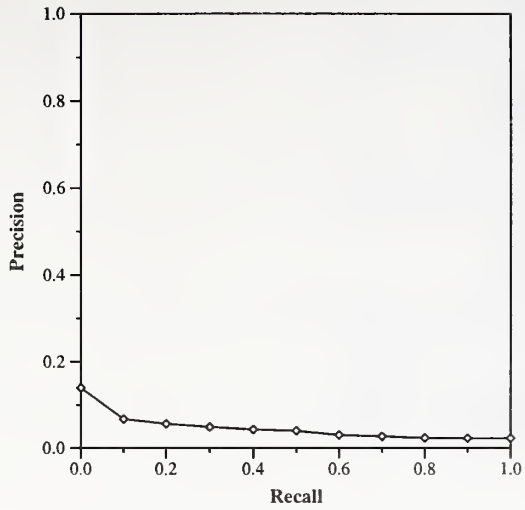


Difference from Median in Average Precision per Topic

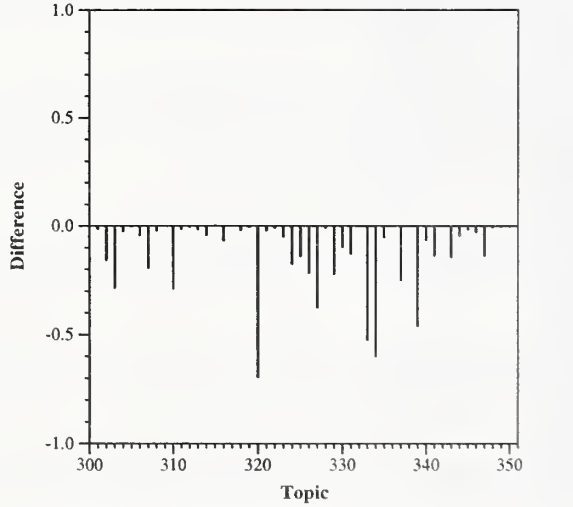
Summary Statistics	
Run Number	Gla6DS3
Run Description	automatic
Number of Topics	47
Total number of documents over all topics	
Retrieved:	47000
Relevant:	1591
Rel-ret:	449

Recall Level Precision Averages	
Recall	Precision
0.00	0.1395
0.10	0.0673
0.20	0.0563
0.30	0.0489
0.40	0.0429
0.50	0.0397
0.60	0.0300
0.70	0.0269
0.80	0.0232
0.90	0.0226
1.00	0.0226
Average precision over all relevant docs	
non-interpolated	0.0416

Document Level Averages	
	Precision
At 5 docs	0.0553
At 10 docs	0.0553
At 15 docs	0.0482
At 20 docs	0.0468
At 30 docs	0.0418
At 100 docs	0.0270
At 200 docs	0.0199
At 500 docs	0.0133
At 1000 docs	0.0096
R-Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.0512



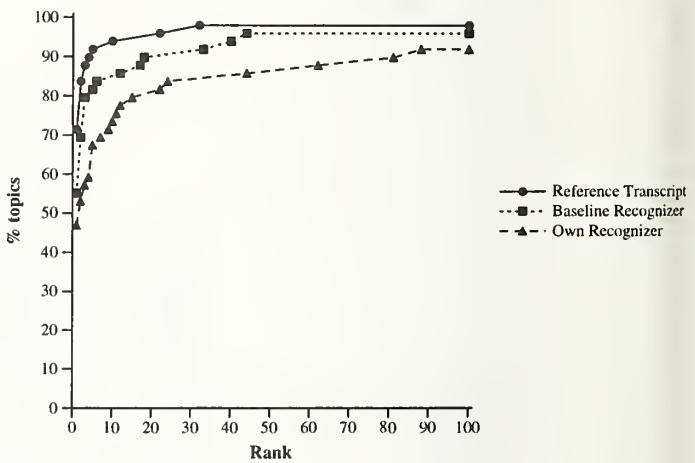
Recall-Precision Curve



Difference from Median in Average Precision per Topic

Table 1: Raw Ranks			
	att97sR1	att97sB1	att97sS1
1	1	1	2
2	5	3	44
3	236	389	157
4	2	1	10
5	1	5	2000
6	1	1	9
7	2	3	3
8	2	6	81
9	2	2	1
10	1	2	1
11	1	1	5
12	4	3	1
13	1	1	5
14	1	1	1
15	1	1	1
16	1	1	1
17	1	1	1
18	1	33	12
19	1	2	5
20	1	2	4
22	1	1	1
23	3	40	157
24	1	1	22
25	1	1	3
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	11
30	32	17	5
31	1	1	1
32	1	1	62
33	1	2	1
34	1	1	1
35	1	1	1
36	22	18	7
37	3	3	1
38	1	3	2
39	1	12	2
40	1	1	1
41	2	1	1
42	1	2000	88
43	1	1	1
44	1	2	1
45	1	1	1
46	1	1	1
47	10	44	24
48	1	1	1
49	2	1	101
50	1	2	15
Mean rank when found	7.39	12.92	17.90
Mean reciprocal rank	0.8020	0.6696	0.5507

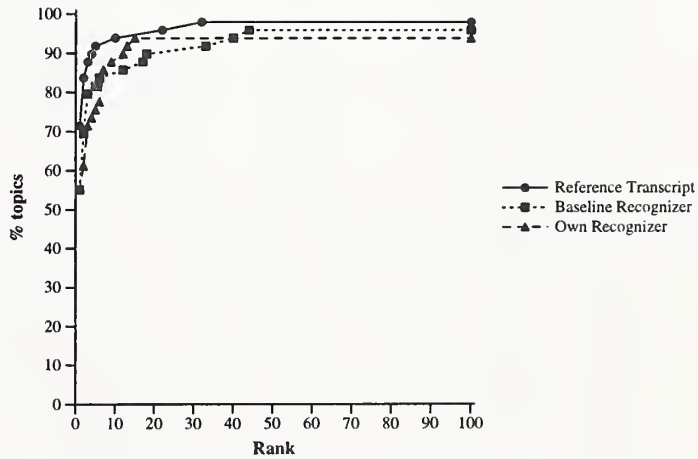
Table 2: Histogram			
Number of items found at rank r where			
	att97sR1	att97sB1	att97sS1
$r \leq 5$	45	40	33
$r \leq 10$	46	41	36
$r \leq 20$	46	44	39
$r \leq 100$	48	47	45
Not found	0	1	1



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	att97sR1	att97sB1	att97sS2
1	1	1	1
2	5	3	3
3	236	389	178
4	2	1	7
5	1	5	3
6	1	1	13
7	2	3	1
8	2	6	6
9	2	2	1
10	1	2	1
11	1	1	7
12	4	3	1
13	1	1	12
14	1	1	1
15	1	1	1
16	1	1	1
17	1	1	1
18	1	33	15
19	1	2	1
20	1	2	2
22	1	1	1
23	3	40	222
24	1	1	7
25	1	1	3
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	1
30	32	17	9
31	1	1	1
32	1	1	4
33	1	2	1
34	1	1	1
35	1	1	1
36	22	18	7
37	3	3	2
38	1	3	3
39	1	12	1
40	1	1	1
41	2	1	1
42	1	2000	116
43	1	1	1
44	1	2	1
45	1	1	1
46	1	1	1
47	10	44	3
48	1	1	1
49	2	1	5
50	1	2	2
Mean rank when found	7.39	12.92	13.39
Mean reciprocal rank	0.8020	0.6696	0.6472

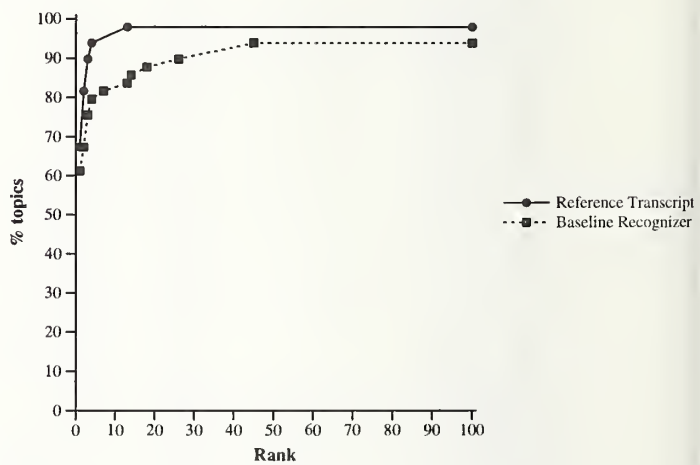
Table 2: Histogram			
Number of items found at rank r where			
	att97sR1	att97sB1	att97sS2
$r \leq 5$	45	40	37
$r \leq 10$	46	41	43
$r \leq 20$	46	44	46
$r \leq 100$	48	47	46
Not found	0	1	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	citysdrR1	citysdrB1	—
1	1	1	
2	3	4	
3	178	161	
4	1	1	
5	1	13	
6	1	1	
7	1	1	
8	4	45	
9	3	3	
10	1	4	
11	2	3	
12	2	3	
13	1	1	
14	1	1	
15	1	1	
16	1	1	
17	1	1	
18	1	18	
19	2	2	
20	1	2	
22	1	1	
23	4	105	
24	2	1	
25	1	1	
26	1	1	
27	1	1	
28	1	1	
29	1	1	
30	13	14	
31	1	1	
32	1	1	
33	2	3	
34	1	1	
35	1	1	
36	1	1	
37	3	1	
38	1	26	
39	3	7	
40	1	1	
41	2	2	
42	2	192	
43	1	1	
44	1	1	
45	1	1	
46	1	1	
47	13	45	
48	1	1	
49	1	1	
50	1	1	
Mean rank when found	5.53	13.92	
Mean reciprocal rank	0.7856	0.6895	

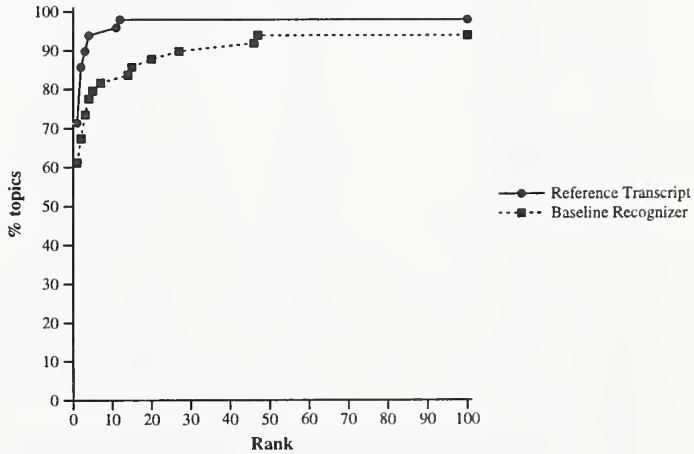
Table 2: Histogram			
Number of items found at rank r where			
	citysdrR1	citysdrB1	—
$r \leq 5$	46	39	
$r \leq 10$	46	40	
$r \leq 20$	48	43	
$r \leq 100$	48	46	
Not found	0	0	



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	citysdrR2	citysdrB2	—
1	1	1	
2	3	4	
3	179	168	
4	1	1	
5	1	15	
6	1	1	
7	1	1	
8	4	47	
9	2	3	
10	1	4	
11	1	5	
12	3	3	
13	1	1	
14	1	1	
15	1	1	
16	1	1	
17	1	1	
18	1	20	
19	2	2	
20	1	2	
22	1	1	
23	4	108	
24	2	1	
25	1	1	
26	1	1	
27	1	1	
28	1	1	
29	1	1	
30	11	14	
31	1	1	
32	1	1	
33	1	3	
34	1	1	
35	1	1	
36	1	1	
37	2	1	
38	1	27	
39	2	7	
40	1	1	
41	2	2	
42	2	186	
43	1	1	
44	1	1	
45	1	1	
46	1	1	
47	12	46	
48	1	1	
49	1	1	
50	1	1	
Mean rank when found	5.41	14.20	
Mean reciprocal rank	0.8132	0.6864	

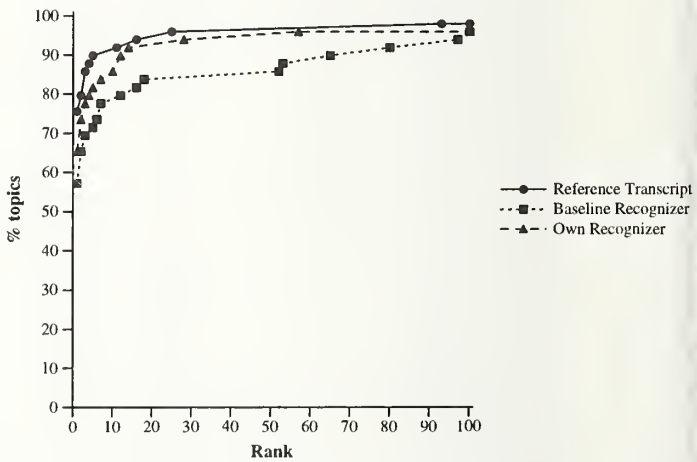
Table 2: Histogram			
Number of items found at rank r where			
	citysdrR2	citysdrB2	—
$r \leq 5$	46	39	
$r \leq 10$	46	40	
$r \leq 20$	48	43	
$r \leq 100$	48	46	
Not found	0	0	



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	CLARITR1	CLARITB1	CLARITS1
1	1	1	1
2	1	1	1
3	123	97	57
4	1	1	2
5	1	3	3
6	2	1	2
7	1	1	1
8	5	18	7
9	4	7	5
10	1	6	2
11	25	108	12
12	1	1	1
13	1	1	1
14	1	1	1
15	1	1	1
16	3	12	4
17	1	1	1
18	1	7	1
19	1	2	1
20	1	3	1
22	1	1	1
23	11	80	28
24	1	1	1
25	1	1	1
26	1	1	1
27	1	2	1
28	1	1	1
29	1	2	1
30	93	541	536
31	1	1	1
32	1	1	1
33	1	1	1
34	1	1	1
35	1	1	1
36	1	1	1
37	1	1	1
38	1	52	1
39	16	65	14
40	1	1	1
41	3	5	3
42	1	100	154
43	1	1	1
44	1	53	2
45	3	2	1
46	1	1	1
47	1	16	12
48	1	1	1
49	2	1	10
50	1	1	1
Mean rank when found	6.67	24.67	18.06
Mean reciprocal rank	0.8094	0.6453	0.7277

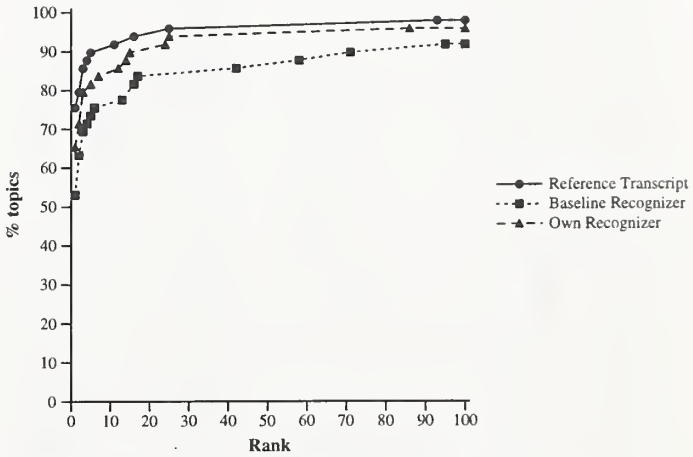
Table 2: Histogram			
Number of items found at rank r where			
	CLARITR1	CLARITB1	CLARITS1
$r \leq 5$	44	35	40
$r \leq 10$	44	38	42
$r \leq 20$	46	41	45
$r \leq 100$	48	47	47
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	CLARITR1	CLARITB2	CLARITS2
1	1	1	1
2	1	1	1
3	123	114	86
4	1	1	2
5	1	3	2
6	2	4	1
7	1	1	1
8	5	13	3
9	4	3	3
10	1	16	3
11	25	111	14
12	1	1	1
13	1	1	1
14	1	1	1
15	1	1	1
16	3	17	5
17	1	1	1
18	1	5	1
19	1	2	1
20	1	3	1
22	1	1	1
23	11	71	24
24	1	1	1
25	1	1	1
26	1	1	1
27	1	2	1
28	1	1	1
29	1	2	1
30	93	693	630
31	1	1	1
32	1	1	1
33	1	1	7
34	1	1	1
35	1	1	1
36	1	1	1
37	1	1	1
38	1	58	1
39	16	95	25
40	1	1	1
41	3	6	3
42	1	123	107
43	1	1	1
44	1	42	2
45	3	2	1
46	1	1	1
47	1	16	12
48	1	1	1
49	2	2	15
50	1	1	1
Mean rank when found	6.67	29.16	19.90
Mean reciprocal rank	0.8094	0.6218	0.7245

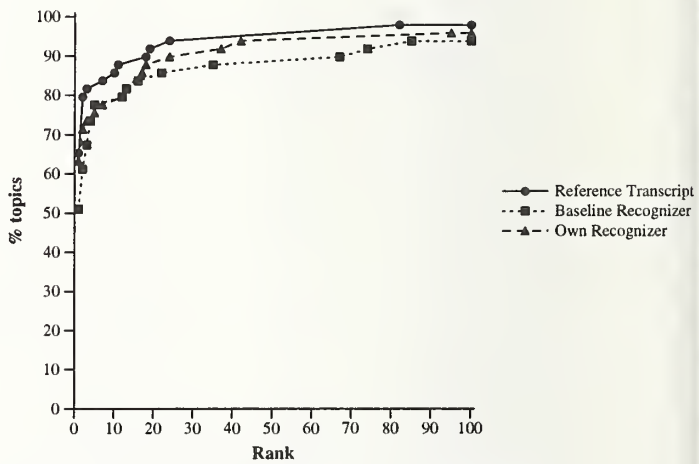
Table 2: Histogram			
Number of items found at rank r where			
	CLARITR1	CLARITB2	CLARITS2
$r \leq 5$	44	36	40
$r \leq 10$	44	37	41
$r \leq 20$	46	41	44
$r \leq 100$	48	45	47
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	CMUref	CMUibm	CMUcmu
1	1	1	1
2	18	4	7
3	435	349	340
4	1	1	1
5	1	3	1
6	82	74	42
7	1	1	1
8	11	85	17
9	10	13	12
10	1	12	13
11	1	2	1
12	2	2	2
13	1	1	1
14	2	2	1
15	1	1	1
16	1	1	1
17	2	2	3
18	1	4	1
19	2	3	1
20	1	2	2
22	1	1	1
23	2	123	37
24	2	1	1
25	1	1	1
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	1
30	82	67	95
31	1	1	2
32	1	1	1
33	1	4	18
34	1	1	1
35	1	1	1
36	3	1	1
37	19	16	17
38	1	22	1
39	1	5	1
40	1	1	1
41	1	1	1
42	24	2000	2000
43	1	1	1
44	1	5	1
45	1	1	1
46	1	1	1
47	7	35	24
48	1	1	1
49	1	1	2
50	2	3	5
Mean rank when found	15.04	17.96	13.94
Mean reciprocal rank	0.7417	0.6122	0.6962

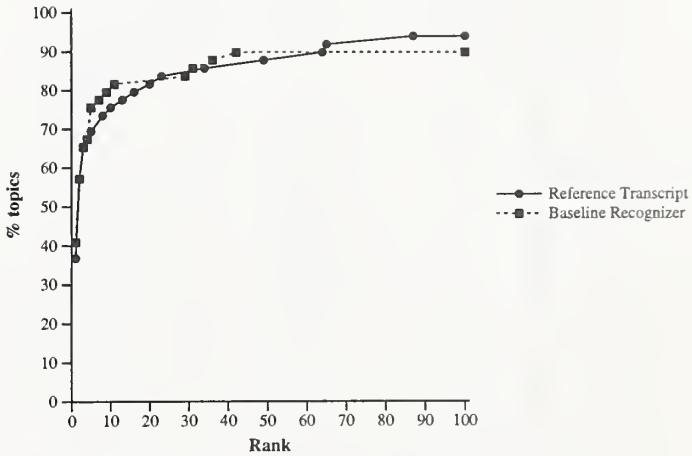
Table 2: Histogram			
Number of items found at rank r where			
	CMUref	CMUibm	CMUcmu
$r \leq 5$	40	38	37
$r \leq 10$	42	38	38
$r \leq 20$	45	41	43
$r \leq 100$	48	46	47
Not found	0	1	1



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	DCUSDRR1	DCUSDRB1	—
1	8	1	
2	34	36	
3	2000	2000	
4	2	1	
5	64	31	
6	1	3	
7	2	3	
8	3	7	
9	23	9	
10	87	2	
11	2000	2000	
12	13	5	
13	2	1	
14	1	1	
15	2	1	
16	1	2000	
17	1	1	
18	1	29	
19	10	5	
20	1	2	
22	3	2	
23	3	294	
24	2	1	
25	5	1	
26	1	1	
27	1	1	
28	1	2	
29	2	1	
30	3	1	
31	1	2	
32	1	1	
33	1	4	
34	1	1	
35	2	2	
36	16	1	
37	1	1	
38	2	11	
39	65	2	
40	2	3	
41	1	5	
42	49	2000	
43	1	1	
44	20	5	
45	2000	1	
46	1	1	
47	8	42	
48	4	2	
49	2	3	
50	1	1	
Mean rank when found	9.91	11.80	
Mean reciprocal rank	0.5196	0.5480	

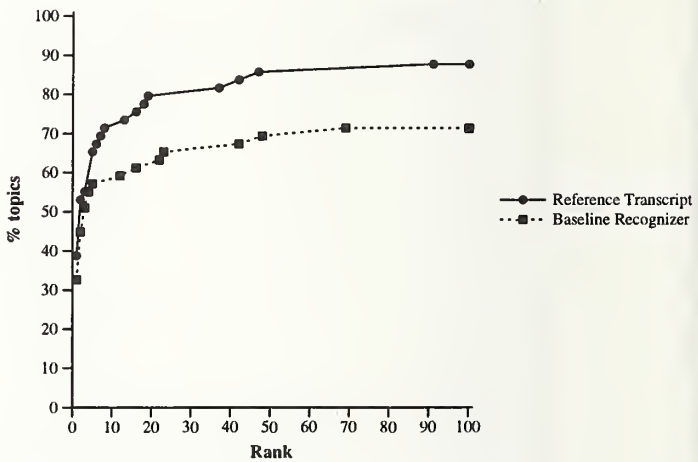
Table 2: Histogram			
Number of items found at rank r where			
	DCUSDRR1	DCUSDRB1	—
$r \leq 5$	34	37	
$r \leq 10$	37	39	
$r \leq 20$	40	40	
$r \leq 100$	46	44	
Not found	3	4	



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	DCUSDRR2	DCUSDRB2	—
1	2	1	
2	18	1	
3	230	2000	
4	13	12	
5	1	2000	
6	5	5	
7	91	1	
8	3	2	
9	16	2000	
10	2000	2000	
11	1	2000	
12	2	2	
13	2	3	
14	2	1	
15	1	2	
16	2000	2000	
17	7	22	
18	1	4	
19	6	42	
20	5	1	
22	1	1	
23	2	48	
24	1	1	
25	1	1	
26	1	1	
27	19	2	
28	5	3	
29	2000	2000	
30	47	23	
31	1	1	
32	2	1	
33	1	1	
34	1	1	
35	5	3	
36	1	2	
37	1	1	
38	2	69	
39	37	2000	
40	1	1	
41	2000	2000	
42	1	2000	
43	1	128	
44	5	2000	
45	1	2000	
46	1	1	
47	8	16	
48	217	2	
49	42	144	
50	1	4	
Mean rank when found	18.04	14.97	
Mean reciprocal rank	0.5022	0.4287	

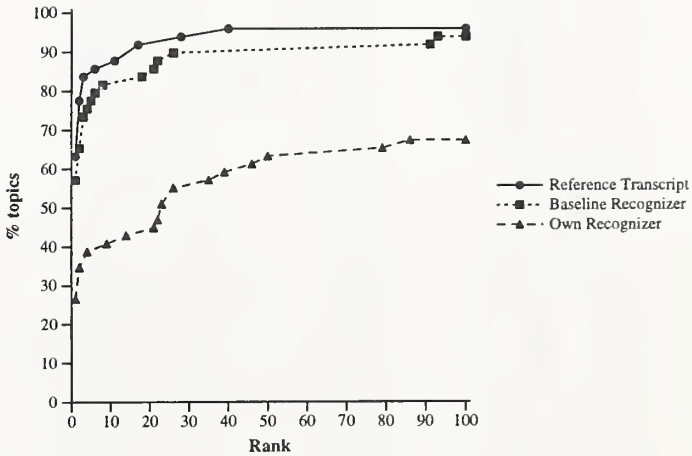
Table 2: Histogram			
Number of items found at rank r where			
	DCUSDRR2	DCUSDRB2	—
$r \leq 5$	32	28	
$r \leq 10$	35	28	
$r \leq 20$	39	30	
$r \leq 100$	43	35	
Not found	4	12	



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	ETHR1	ETHB1	ETHS1
1	1	1	21
2	1	1	182
3	218	371	716
4	1	1	1
5	1	4	1
6	1	1	79
7	1	1	1
8	3	5	151
9	17	18	505
10	2	2	158
11	1	3	4
12	3	2	4
13	6	1	1
14	1	3	39
15	1	1	2
16	28	26	400
17	1	1	1
18	1	1	120
19	1	6	195
20	1	1	443
22	1	1	1
23	11	91	776
24	1	1	1
25	2	1	2
26	1	1	26
27	3	3	1
28	1	2	2
29	2	2	659
30	481	432	474
31	1	1	1
32	1	1	1
33	1	1	35
34	1	1	22
35	1	1	23
36	1	1	9
37	1	1	86
38	1	22	46
39	2	8	14
40	1	1	1
41	1	1	295
42	17	575	585
43	1	1	50
44	1	93	2
45	1	1	23
46	2	1	1
47	2	21	26
48	1	1	229
49	40	1	225
50	2	3	1
Mean rank when found	17.80	35.10	135.53
Mean reciprocal rank	0.7335	0.6590	0.3290

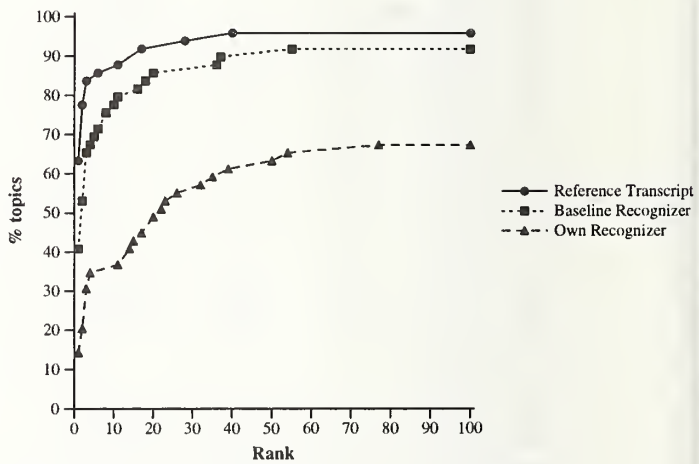
Table 2: Histogram			
Number of items found at rank r where			
	ETHR1	ETHB1	ETHS1
$r \leq 5$	41	38	19
$r \leq 10$	42	40	20
$r \leq 20$	45	41	21
$r \leq 100$	47	46	33
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	ETHR1	ETHB2	ETHS2
1	1	1	20
2	1	1	185
3	218	371	716
4	1	1	1
5	1	10	1
6	1	2	39
7	1	4	2
8	3	8	156
9	17	18	505
10	2	3	182
11	1	3	3
12	3	5	14
13	6	1	1
14	1	8	54
15	1	1	11
16	28	36	407
17	1	3	4
18	1	2	113
19	1	3	215
20	1	3	443
22	1	1	1
23	11	105	758
24	1	1	2
25	2	1	2
26	1	1	15
27	3	6	3
28	1	1	4
29	2	2	659
30	481	432	466
31	1	1	1
32	1	1	1
33	1	11	14
34	1	2	22
35	1	1	20
36	1	1	23
37	1	3	77
38	1	20	35
39	2	16	17
40	1	1	3
41	1	2	295
42	17	575	585
43	1	1	50
44	1	55	3
45	1	2	26
46	2	1	1
47	2	37	32
48	1	1	167
49	40	1	230
50	2	1	3
Mean rank when found	17.80	36.06	134.43
Mean reciprocal rank	0.7335	0.5370	0.2337

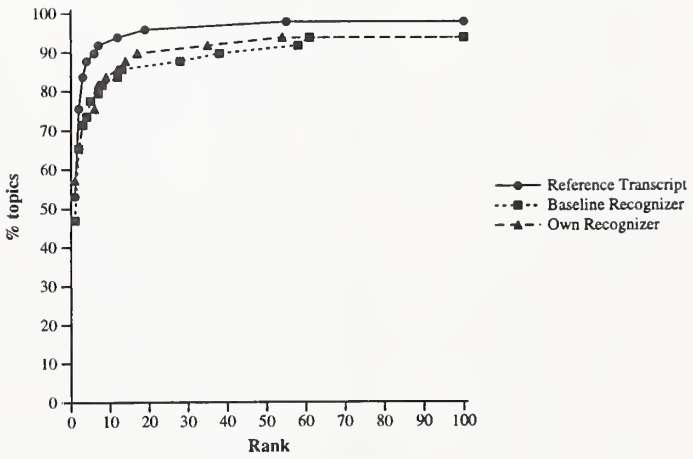
Table 2: Histogram			
Number of items found at rank r where			
	ETHR1	ETHB2	ETHS2
$r \leq 5$	41	34	17
$r \leq 10$	42	38	17
$r \leq 20$	45	42	24
$r \leq 100$	47	45	33
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	gla6R1	gla6B1	gla6S1
1	1	1	1
2	7	4	7
3	227	200	240
4	1	1	1
5	1	7	14
6	1	2	1
7	1	2	1
8	6	28	7
9	4	2	12
10	1	13	3
11	1	2	1
12	2	2	3
13	1	1	1
14	1	2	2
15	1	1	1
16	2	1	1
17	2	3	2
18	1	5	3
19	2	2	1
20	1	2	1
22	1	1	1
23	3	110	329
24	2	1	1
25	2	1	2
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	1
30	55	61	35
31	2	1	1
32	1	1	1
33	3	12	2
34	1	1	1
35	1	1	1
36	4	2	1
37	12	3	6
38	3	38	7
39	2	5	6
40	3	1	1
41	2	1	1
42	2	273	264
43	1	1	1
44	1	3	1
45	1	1	1
46	1	1	1
47	19	58	54
48	1	1	1
49	2	8	17
50	1	1	9
Mean rank when found	8.04	17.80	21.47
Mean reciprocal rank	0.6898	0.6059	0.6560

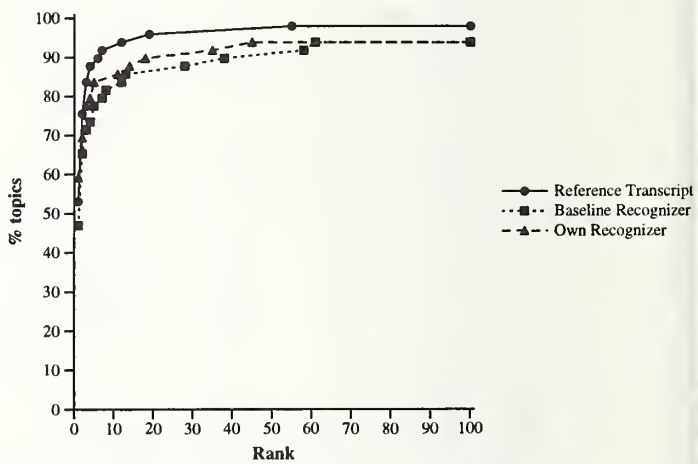
Table 2: Histogram			
Number of items found at rank r where			
	gla6R1	gla6B1	gla6S1
$r \leq 5$	43	38	35
$r \leq 10$	45	40	41
$r \leq 20$	47	42	44
$r \leq 100$	48	46	46
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	gla6R1	gla6B1	gla6S2
1	1	1	1
2	7	4	3
3	227	200	196
4	1	1	1
5	1	7	11
6	1	2	1
7	1	2	1
8	6	28	14
9	4	2	3
10	1	13	1
11	1	2	2
12	2	2	2
13	1	1	1
14	1	2	1
15	1	1	1
16	2	1	1
17	2	3	1
18	1	5	4
19	2	2	2
20	1	2	2
22	1	1	1
23	3	110	167
24	2	1	1
25	2	1	1
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	1
30	55	61	35
31	2	1	1
32	1	1	1
33	3	12	2
34	1	1	1
35	1	1	1
36	4	2	1
37	12	3	3
38	3	38	18
39	2	5	5
40	3	1	1
41	2	1	1
42	2	273	279
43	1	1	1
44	1	3	1
45	1	1	1
46	1	1	1
47	19	58	45
48	1	1	1
49	2	8	5
50	1	1	3
Mean rank when found	8.04	17.80	16.94
Mean reciprocal rank	0.6898	0.6059	0.6891

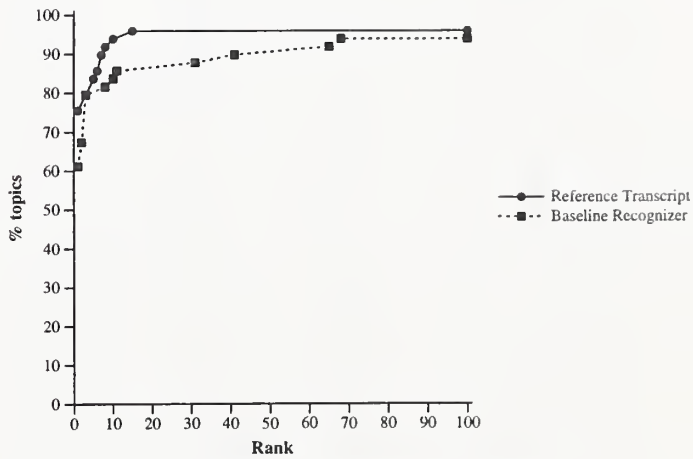
Table 2: Histogram			
Number of items found at rank r where			
	gla6R1	gla6B1	gla6S2
$r \leq 5$	43	38	41
$r \leq 10$	45	40	41
$r \leq 20$	47	42	44
$r \leq 100$	48	46	46
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	ibms97t	ibms97s	—
1	1	1	
2	5	1	
3	331	352	
4	1	1	
5	1	2	
6	1	1	
7	1	3	
8	7	41	
9	5	10	
10	1	1	
11	1	1	
12	7	8	
13	1	1	
14	1	1	
15	1	1	
16	1	1	
17	1	1	
18	1	68	
19	1	2	
20	1	1	
22	1	1	
23	3	65	
24	1	1	
25	1	1	
26	1	1	
27	1	3	
28	1	1	
29	1	1	
30	143	321	
31	1	1	
32	1	1	
33	1	3	
34	1	1	
35	1	1	
36	6	3	
37	8	11	
38	1	2	
39	1	3	
40	1	1	
41	3	1	
42	15	524	
43	1	1	
44	1	3	
45	1	1	
46	1	1	
47	10	31	
48	1	1	
49	1	1	
50	1	1	
Mean rank when found	11.84	30.31	
Mean reciprocal rank	0.7923	0.6921	

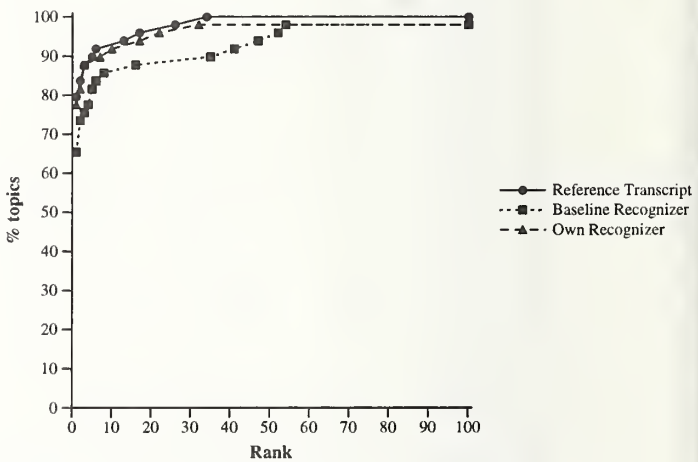
Table 2: Histogram			
Number of items found at rank r where			
	ibms97t	ibms97s	—
$r \leq 5$	41	39	
$r \leq 10$	46	41	
$r \leq 20$	47	42	
$r \leq 100$	47	46	
Not found	0	0	



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	INQ4sdl	INQ4sds	INQ4sdd
1	1	1	1
2	1	1	1
3	26	41	201
4	1	1	2
5	1	2	1
6	1	5	1
7	1	1	1
8	6	35	10
9	1	1	3
10	1	1	1
11	1	1	1
12	17	16	17
13	1	1	1
14	1	1	1
15	1	1	1
16	1	1	1
17	1	1	1
18	1	6	1
19	1	5	2
20	1	2	1
22	1	1	1
23	3	52	3
24	1	1	1
25	1	1	1
26	1	1	1
27	2	1	1
28	1	1	1
29	1	1	1
30	34	47	1
31	1	1	1
32	1	1	1
33	1	8	1
34	1	1	1
35	1	1	1
36	5	3	3
37	1	1	1
38	1	2	1
39	1	4	1
40	1	1	1
41	1	1	1
42	3	308	22
43	1	1	1
44	1	2	1
45	1	1	1
46	1	1	1
47	13	54	32
48	1	1	1
49	2	1	7
50	1	1	1
Mean rank when found	3.06	12.73	6.94
Mean reciprocal rank	0.8416	0.7235	0.8242

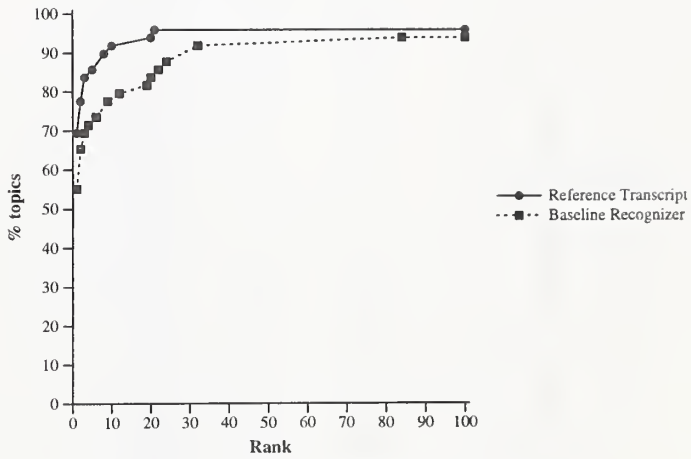
Table 2: Histogram			
Number of items found at rank r where			
	INQ4sdl	INQ4sds	INQ4sdd
$r \leq 5$	44	40	43
$r \leq 10$	45	42	45
$r \leq 20$	47	43	46
$r \leq 100$	49	48	48
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	nsasglt1	nsasgsr1	—
1	1	1	
2	1	1	
3	550	493	
4	1	2	
5	1	20	
6	1	1	
7	1	6	
8	8	84	
9	5	3	
10	1	1	
11	1	1	
12	20	24	
13	1	1	
14	1	12	
15	1	1	
16	2	2	
17	1	1	
18	1	1	
19	1	2	
20	1	2	
22	1	1	
23	3	22	
24	1	1	
25	2	1	
26	1	1	
27	21	19	
28	1	1	
29	1	1	
30	215	220	
31	1	1	
32	1	1	
33	1	1	
34	10	9	
35	1	1	
36	8	9	
37	1	1	
38	3	32	
39	1	4	
40	1	1	
41	2	2	
42	1	314	
43	1	1	
44	1	3	
45	1	1	
46	1	1	
47	3	32	
48	1	1	
49	2	1	
50	1	1	
Mean rank when found	18.12	27.41	
Mean reciprocal rank	0.7685	0.6360	

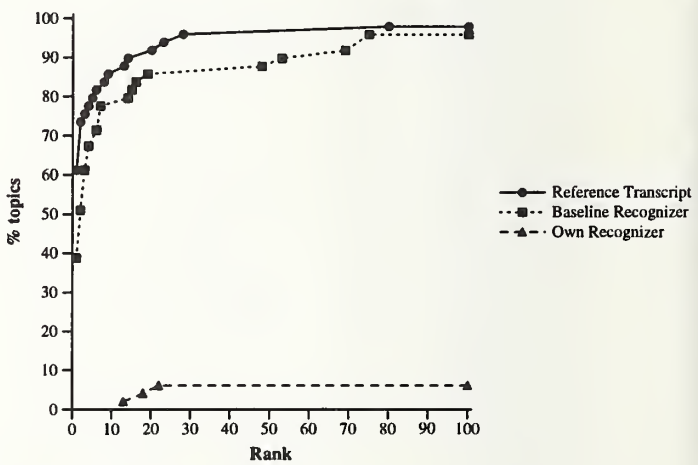
Table 2: Histogram			
Number of items found at rank r where			
	nsasglt1	nsasgsr1	—
$r \leq 5$	42	35	
$r \leq 10$	45	38	
$r \leq 20$	46	41	
$r \leq 100$	47	46	
Not found	0	0	



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	mds612	mds613	mds614
1	1	1	13
2	1	1	2000
3	2000	2000	319
4	9	6	171
5	1	16	2000
6	1	3	18
7	1	2	163
8	13	75	2000
9	14	14	2000
10	1	3	2000
11	1	1	2000
12	20	15	162
13	2	2	448
14	1	1	22
15	1	1	2000
16	1	1	2000
17	1	1	2000
18	1	53	2000
19	1	4	2000
20	1	3	2000
22	1	1	2000
23	4	75	2000
24	1	2	2000
25	3	3	2000
26	1	1	2000
27	8	7	2000
28	1	1	2000
29	1	1	283
30	28	19	2000
31	2	1	2000
32	1	1	172
33	1	7	2000
34	2	2	288
35	1	3	2000
36	6	2	2000
37	80	69	2000
38	1	7	2000
39	2	4	2000
40	1	1	2000
41	1	1	2000
42	23	2000	2000
43	1	1	2000
44	1	1	2000
45	1	1	2000
46	2	2	381
47	5	48	333
48	1	1	425
49	1	4	2000
50	2	6	240
Mean rank when found	5.31	10.11	229.20
Mean reciprocal rank	0.7036	0.5207	0.0046

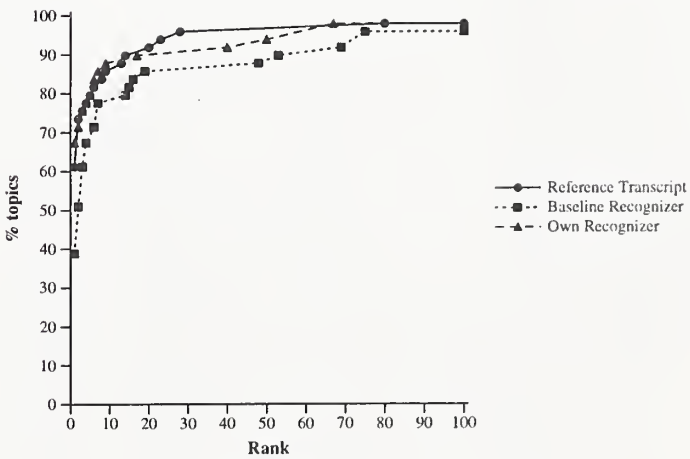
Table 2: Histogram			
Number of items found at rank r where			
	mds612	mds613	mds614
$r \leq 5$	39	33	
$r \leq 10$	42	38	
$r \leq 20$	45	42	2
$r \leq 100$	48	47	3
Not found	1	2	34



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	mds612	mds613	mds615
1	1	1	1
2	1	1	1
3	2000	2000	106
4	9	6	5
5	1	16	1
6	1	3	1
7	1	2	1
8	13	75	50
9	14	14	6
10	1	3	1
11	1	1	1
12	20	15	4
13	2	2	1
14	1	1	1
15	1	1	1
16	1	1	17
17	1	1	40
18	1	53	1
19	1	4	1
20	1	3	1
22	1	1	1
23	4	75	3
24	1	2	1
25	3	3	2
26	1	1	1
27	8	7	2
28	1	1	1
29	1	1	1
30	28	19	67
31	2	1	1
32	1	1	1
33	1	7	1
34	2	2	1
35	1	3	1
36	6	2	67
37	80	69	7
38	1	7	1
39	2	4	1
40	1	1	1
41	1	1	1
42	23	2000	3
43	1	1	1
44	1	1	1
45	1	1	1
46	2	2	1
47	5	48	9
48	1	1	1
49	1	4	1
50	2	6	6
Mean rank when found	5.31	10.11	8.71
Mean reciprocal rank	0.7036	0.5207	0.7316

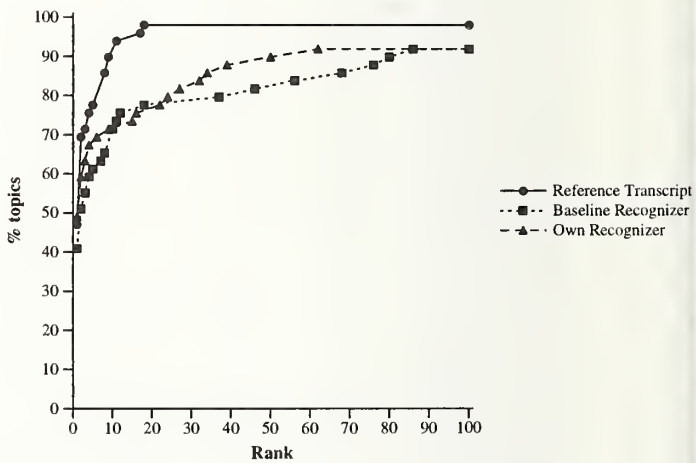
Table 2: Histogram			
Number of items found at rank r where			
	mds612	mds613	mds615
$r \leq 5$	39	33	39
$r \leq 10$	42	38	43
$r \leq 20$	45	42	44
$r \leq 100$	48	47	48
Not found	1	2	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	THISLR1	THISLB1	THISLS1
1	1	1	1
2	2	3	3
3	400	349	372
4	1	1	1
5	2	86	125
6	1	10	1
7	1	2	2
8	18	37	32
9	8	8	22
10	2	76	34
11	5	135	39
12	2	4	4
13	1	1	1
14	2	3	2
15	1	1	1
16	8	7	2
17	2	2	1
18	1	10	4
19	2	2	1
20	1	2	1
22	1	1	1
23	11	160	268
24	2	1	1
25	1	1	1
26	1	1	1
27	1	2	2
28	1	1	1
29	3	5	1
30	9	68	62
31	1	1	1
32	1	1	1
33	8	46	6
34	1	1	1
35	1	1	1
36	1	1	1
37	11	12	15
38	8	80	27
39	4	11	9
40	4	4	3
41	2	1	1
42	9	263	214
43	1	1	1
44	1	18	1
45	2	1	2
46	1	1	1
47	17	56	50
48	1	1	1
49	1	10	24
50	2	1	16
Mean rank when found	11.59	30.43	27.82
Mean reciprocal rank	0.6236	0.5062	0.5784

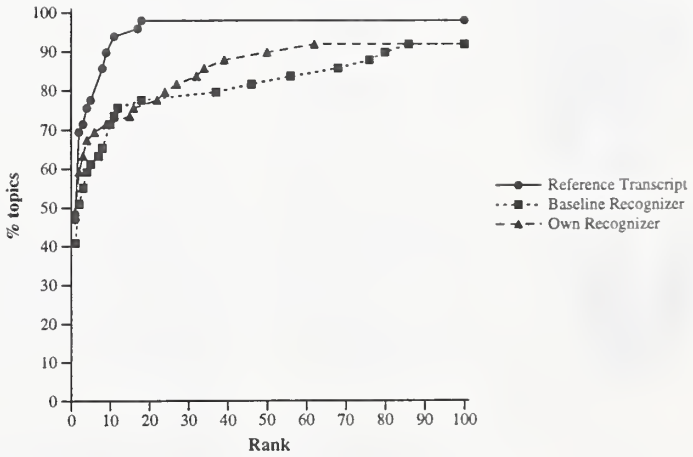
Table 2: Histogram			
Number of items found at rank r where			
	THISLR1	THISLB1	THISLS1
$r \leq 5$	38	30	33
$r \leq 10$	44	35	35
$r \leq 20$	48	38	37
$r \leq 100$	48	45	45
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	THISLR2	THISLB2	THISLS2
1	1	1	1
2	2	3	3
3	400	349	372
4	1	1	1
5	2	86	125
6	1	10	1
7	1	2	2
8	18	37	32
9	8	8	22
10	2	76	34
11	5	135	39
12	2	4	4
13	1	1	1
14	2	3	2
15	1	1	1
16	8	7	2
17	2	2	1
18	1	10	4
19	2	2	1
20	1	2	1
22	1	1	1
23	11	160	268
24	2	1	1
25	1	1	1
26	1	1	1
27	1	2	2
28	1	1	1
29	3	5	1
30	9	68	62
31	1	1	1
32	1	1	1
33	8	46	6
34	1	1	1
35	1	1	1
36	1	1	1
37	11	12	15
38	8	80	27
39	4	11	9
40	4	4	3
41	2	1	1
42	9	263	214
43	1	1	1
44	1	18	1
45	2	1	2
46	1	1	1
47	17	56	50
48	1	1	1
49	1	10	24
50	2	1	16
Mean rank when found	11.59	30.43	27.82
Mean reciprocal rank	0.6236	0.5062	0.5784

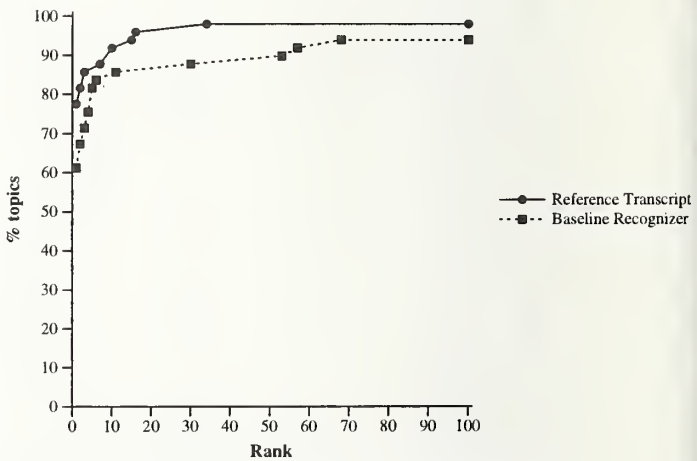
Table 2: Histogram			
Number of items found at rank r where			
	THISLR2	THISLB2	THISLS2
$r \leq 5$	38	30	33
$r \leq 10$	44	35	35
$r \leq 20$	48	38	37
$r \leq 100$	48	45	45
Not found	0	0	0



Cumulative % of topics that retrieve target item by given rank

Table 1: Raw Ranks			
	umcp97l1	umcp97s2	—
1	1	1	
2	1	1	
3	237	183	
4	1	1	
5	1	5	
6	1	1	
7	1	1	
8	10	125	
9	1	1	
10	1	1	
11	1	1	
12	10	11	
13	1	1	
14	1	1	
15	1	1	
16	1	1	
17	1	1	
18	1	53	
19	1	1	
20	1	2	
22	1	1	
23	3	68	
24	1	1	
25	3	1	
26	1	1	
27	2	4	
28	1	1	
29	1	1	
30	34	30	
31	1	1	
32	1	1	
33	1	4	
34	1	1	
35	1	1	
36	7	5	
37	15	6	
38	1	5	
39	1	2	
40	1	1	
41	1	1	
42	1	283	
43	1	1	
44	1	2	
45	1	1	
46	1	1	
47	16	57	
48	1	1	
49	2	3	
50	1	3	
Mean rank when found	7.69	17.98	
Mean reciprocal rank	0.8198	0.6862	

Table 2: Histogram			
Number of items found at rank r where			
	umcp97l1	umcp97s2	—
$r \leq 5$	42	40	
$r \leq 10$	45	41	
$r \leq 20$	47	42	
$r \leq 100$	48	46	
Not found	0	0	



Cumulative % of topics that retrieve target item by given rank

SUMMARY PERFORMANCE COMPARISONS TREC-2, TREC-3, TREC-4, TREC-5, TREC-6

Karen Sparck Jones
Computer Laboratory, University of Cambridge

December 20, 1997

Data

These comparisons are designed to allow trends over successive TRECs to emerge. But they are selective views, and do not summarise all there is to say about TREC overall. (TREC-1 is not included as it was a start-up, debugging, cycle.)

The performance comparisons between TREC-2, TREC-3, TREC-4 and TREC-5 originally appeared as Appendix B to the TREC-5 Proceedings, i.e. to *The Fifth Text REtrieval Conference (TREC-5)*, Ed. E.M. Voorhees and D.K. Harman, NIST Special Publication 500-238, National Institute of Standards and Technology, 1997 (and see also the Editors' opening Overview).

The TREC-6 figures, like those for TREC-4 and TREC-5, are from the relevant conference working papers.

The tables cover Adhoc and Routing task results respectively. The Adhoc task, originally mandatory, has later still been strongly recommended, and most participants have therefore submitted runs for it. Routing has become more optional, but has still attracted submissions, and figures for it for TREC-6, its final year, are therefore included for completeness. Other tasks, i.e. tracks, are not covered here since, though they have become more important for TREC as a whole, they are too variable in definition and participation for systematic comparisons.

In the earlier comparisons for TREC-2 - 4, changes in the nature of the topics used in successive TRECs were disregarded, and no distinctions were made between automatic and manual search queries. However for the Adhoc case in particular, there were further changes in TREC-5 and in TREC-6 in both the character of the supplied topic data and the precise specification of the options within the task. Trend comparisons over the whole series of TRECs can therefore only be at a general level, rather than in detail.

Thus for Adhoc up to TREC-4, the quantity and quality of content supplied per topic was reduced. But automatic and manual *modes* of query formation were accepted as legitimate alternatives. So in the comparisons for TREC-2 - 4 the best performing of the official runs submitted per team have been used, regardless of whether this was obtained by automatic or manual processing (details can be recovered from the Proceedings). However for TREC-5, the Adhoc topics were 'split' to give two *versions*, Short (S) and Long (L), with the latter adding content to and subsuming the former. S was obligatory for automatic searching. L was supplied for manual searching, which could also be rather more interactive than in previous TRECs, and was also optional for automatic. For TREC-6, the topics were further separated into Very short (V), Short and Long, with automatic runs

on S required and V and L runs optional. But it should be noted that it was found that in the original topic preparation for TREC-6 the S versions did not necessarily subsume the V ones: the S versions not infrequently covered only addditional, complementary topic content (see Voorhees and Harman's Overview). This probably accounts for the relatively good V performance and relatively poor S results. Manual searching was taken as using the L versions, as for TREC-5, and could be interactive. The comparisons for TREC-5 and TREC-6 Adhoc cover the best runs for each team for each topic version and query mode.

The data and relevant run sets used for the main comparison tables below are therefore as follows:

Adhoc

topic components	TREC 1	2	3	4	5			6	
					S	L	V	S	L
title	x	x	x	x		x	x		x
description	x	x	x	x	x	x		x	x
narrative	x	x	x			x			x
concepts	x	x							
queries	a/m	a/m	a/m	a/m	a	m/(a)	(a)	a	m/(a)

Routing

topics for TREC-1 - 3 were in the same style as Adhoc TREC-1 - 2
topics for TREC-4 - 6 were subsets drawn from all previous topics, so had variable composition.

Further comments and analysis follow the main tables.

Tables 1 and 2 present Precision performance at Document Cutoff 30. Table 1, for Adhoc, is divided into results for TREC-2 - 4 (Table 1a) and for TREC-5 - 6 (1b).

The data are only for full Category A runs, not Category B, and cover only the higher levels of performance, not all the submitted runs.

The conventions are as follows: figures are not rounded; performance is assigned to 'blocks'; teams per block are NOT in merit order, but in original run list order; the best of two official runs is taken, regardless of the particular strategy used, where there are two and these are deemed legitimate alternatives. Simple, hopefully sufficiently identifiable, short names have been given to the teams.

TABLE 1a

ADHOC - DOCUMENT CUTOFF 30

	TREC-2	TREC-3	TREC-4	==> (TREC-5, -6 Table b)
	a/m	a/m	a/m	
>= 60		UMass		
		City		
		Berkeley		
>= 55	UMass	Cornell		
	HNC	Mead		
	VT			
>= 50	Cornell	Verity		
	Berkeley	VT		
	Dortmund	Westlaw		
	CMU/Clarit	ETH		
	Verity	CUNY		
	Siemens			
	CUNY			
>= 45	City	NYU	Excalibur/	==> (best TREC-5 level)
	Bellcore	CMU/Clarit	Conquest	
	ETH	RMIT	CUNY	
	CITRI/RMIT	RutgersK	Waterloo	
	Conquest			
>= 40	...	...	Berkeley	
			Clarit/CMU	
			Cornell	
			GMU	
			UMass	
			InText	
			ANU	
>= 35	...	...	City	
			GE/NYU	
>= 30	...	...	...	
>= 25	...	...	...	
>= 20	...	...	...	==>

TABLE 1b

ADHOC - DOCUMENT CUTOFF 30

	TREC-5 S a	TREC-5 L a	TREC-5 L m	TREC-6 V a	TREC-6 S a	TREC-6 L a	TREC-6 L m
>= 60							
>= 55							
>= 50							Waterloo
>= 45			ETH				Clarit
>= 40			Waterloo				ANU
>= 35	Lexis *		ANU Clarit Cornell GE/NYU GMUetc Lexis				GEetc Lexis
>= 30		City CUNY ETH	OpenText CUNY Berkeley	Apple ATT City IRIT Lexis CUNY Waterloo		ANU Cornell IRIT CUNY Berkeley	ISS Berkeley
>= 25	Apple City Cornell IBMTJW ETH UMass	Apple GE/NYU RMIT Berkeley	DCU IBM	DCU ISS	ATT ANU City Cornell GMUetc IBMTJWs IRIT Lexis Waterloo	City IBMTJWg MDS/RMIT UMass GMUetc	FS GMUetc
>= 20	...	...	...	MDS/RMIT Glasgow	Apple GEetc IBMTJWg MDS/RMIT CUNY Berkeley Maryland UMass Verity	Verity	Glasgow

* reclassified to manual after the conference

TABLE 2

ROUTING - DOCUMENT CUTOFF 30

	TREC-2	TREC-3	TREC-4	TREC-5	TREC-6
>= 60	Cornell Dortmund	City			
>= 55	City Berkeley UMass Bellcore CMU/Clarit CUNY	UMass Cornell Berkeley Dortmund Bellcore	City UMass Xerox		ATT
>= 50	Rutgers HNC GE TRW Verity Siemens	CMU/Clarit Westlaw Logicon TRW Florida	Cornell CUNY	City Cornell UMass	City Cornell CUNY
>= 45	VT	Xerox NYU Verity ETH NSA NEC	Logicon GE/NYU	CUNY	Clarit IRIT ETH Waterloo
>= 40	...	...	...	GE/NYU ETH Berkeley	SRI Berkeley UCSD UMass Verity

Comments

For both Adhoc and Routing, constant participants who initially performed well have continued to do so, though they have sometimes not profited from experiments, while others who started less well have successfully improved their performance. However many teams have not participated throughout the series, or have more recently concentrated on the tracks, so no inferences should be drawn where teams figure only occasionally in the tables.

Adhoc

1. Ad hoc best performance improved from TREC-2 to TREC-3, even though the TREC-3 topics were less rich. The sharp fall in performance for TREC-4 must reflect the minimal topics given for the tests. The further decline in TREC-5, even for the L topics which were fuller than the TREC-4 topics and like the TREC-3 ones, reflects the fact that the topics were deemed 'difficult' in relation to the definition of relevant documents. Performance for TREC-5 and TREC-6 is generally similar, presumably reflecting a data 'plateau', even for the L versions of the topics.

2. The lower levels of performance (even for the better-performing teams) in TREC-4 - TREC-6 must be taken as representing a more realistic retrieval situation than TREC-2 and TREC-3. This statement has, however, to be heavily qualified. The L versions of the topics used in TREC-5 and TREC-6, though less elaborate than the very full earlier ones, are still more elaborate than are typically encountered in Adhoc retrieval practice, especially as end-user input to an automatic system. The defects of the TREC-6 S versions, already noted, probably depressed performance, and the S results for TREC-4 and TREC-5 may have been similarly affected. It is thus unfortunately not possible to draw any grounded inferences, based on systematic comparisons, about the effects of increasing topic fullness on performance. The only runs with any fairly direct bearing on practical situations, where end-users approach automatic search systems in a simple-minded way, are therefore only the (optional) TREC-6 V version ones. These suggest that where at least some attention is paid to the choice of the few initial search terms, adequate, though not high, performance can be obtained with automatic techniques, even without explicit relevance feedback. The V versions averaged only 2.6 words per topic.

   (It would be very useful to enhance the TREC-6 results with new 'proper' short version runs: these experiments would use topic titles as well as descriptions, and thus be true intermediates between V and L: to avoid confusion with the older Short versions, they might be labelled "Medium".)

3. In TREC-2 and TREC-3 automatic query formation was more common than manual, and often performed well, appearing even in the top blocks. Indeed there was relatively more use of automatic query in TREC-3 than TREC-2. But in TREC-4 there was a clear shift towards manual, doubtless in response to the perceived need to beef up the initial minimal topics, with almost all the teams covered by the table using manual queries. However at least one of the top-level teams using automatic query (Cornell) continued to do comparatively well in TREC-4. It is evident that manual query formation was advantageous for TREC-5 even when the same, quite full, initial topic information (L) was used, and the same applies to TREC-6.

But it is important to note that the definition "manual" covers a wide range of human effort from the fairly minimal to the very intensive, and was also explicitly widened in TREC-5 to allow 'feedback' strategies.It is nevertheless not clear, in general, what forms of manual device or effort are especially profitable, or how far intensive effort (and hence time) pays off, or how manual input and automatic devices are best combined. In earlier TRECs it appeared that relatively modest human effort could deliver as well as much more intensive work, but this was from good bases. Detailed analysis is needed for TREC-4 - 6

Routing

1. Routing has shown a slow decline in overall performance, again reflecting less good topic starting information: thus both TREC-4 and TREC-5 had 'tough' topics. Performance in TREC-6 is slightly better, reflecting data with better fitting between topics and document file.

2. In general, when topics are not problematic, and there is rich training data (as with TREC-2 and TREC-3 and also TREC-6) a good level of performance can be obtained.

3. In TREC-2 automatic query formation was only slightly more common than manual, but by TREC-5 manual formation (for the teams shown in the table) had disappeared, reflecting the value of the large training data availability and its utility in compensating for any weakness in automatic query management.

Comparing Adhoc and Routing results, performance in TREC-2 and TREC-3 reached similar levels, attributable to the good topic specifications for the former even though Routing also benefitted in the same way and from the training data. However Adhoc performance has fallen much below that for Routing in TREC-4 and TREC-5, since Routing has been able to benefit from training data.

Overall remarks

1. Many (very) different approaches give similar performance.

2. The general findings about retrieval strategies for early TRECs reported in 'Reflections on TREC', *Information Processing and Management* 31 (3), 1995, p 309 and p 311 respectively, still essentially hold. Thus term weighting, query expansion and so forth are valuable, and in automatic searching quite simple strategies can be as effective as more elaborated ones, so e.g. sophisticated natural language processing is not especially helpful. This has led to some convergence on what may be called the *generic* $tf * idf$ paradigm with relevance feedback refinement. But even with good data (as illustrated by the TREC-6 L version topics, Precision at Document Cutoff 30 is more often than not below 30 %. For the collection data this corresponds roughly to Recall of 30 %.

3. Moreover the range of specific devices, and of combinations of devices, in TREC remains very wide, so more understanding of the effects of environment variables on system parameters for large text files is required, while, as already noted, a detailed comparative analysis of what manual query formation contributes would be very useful.

4. All the points made here are broad brush ones, and the nature of the tables must always be borne in mind. Thus there may be real differences within performance blocks, and also none between members of adjoining blocks. However as it is not obvious what performance differences are statistically significant, what differences are meaningful to users, and whether significant differences are also meaningful, it is only proper to take a generally rather conservative view of apparent performance differences in the tables. More concretely, Precision of 45 % and 35 % are respectively equivalent to 13.5 and 10.5 relevant documents retrieved, a difference which may not matter much to a user; and even if the difference between 45 % and 40 % was statistically significant, the corresponding difference between 13.5 and 12 relevant documents retrieved would almost certainly not matter.

☆U.S. GOVERNMENT PRINTING OFFICE:1998-433-235/93225

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bimonthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program in support of the efforts of private-sector standardizing organizations.

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency or Internal Reports (NISTIR)—The series includes interim or final reports on work performed by NIST for outside sponsors (both government and nongovernment). In general, initial distribution is handled by the sponsor; public distribution is handled by sales through the National Technical Information Service, Springfield, VA 22161, in hard copy, electronic media, or microfiche form. NISTIR's may also report results of NIST projects of transitory or limited interest, including those that will be published subsequently in more comprehensive form.

U.S. Department of Commerce
National Institute of Standards
and Technology
Gaithersburg, MD 20899–0001