

UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a master's thesis by

YUNJIANG LUO

and have found that it is completed and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

CAROLYN J. CROUCH

Name of Faculty Advisor

Signature of Faculty Advisor

Date

GRADUATE SCHOOL

QUERY REFORMULATION IN TREC

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA

BY

YUNJIANG LUO

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

JUNE 1997

Acknowledgements

I wish to express my sincere and deep thanks to my thesis advisor, Professor Carolyn Crouch. Without her guidance and encouragement I cannot finish the research.

I am grateful to Professor Donald Crouch for taking time to read the thesis and make comments and suggestions.

I also wish to extend my personal gratitude to Professor Harlan Stech, for taking time and interest to serve on my committee. I am also thankful to all the faculty members and the graduate students of the Department of Computer Science, University of Minnesota, Duluth.

ABSTRACT

An information retrieval system based on the vector space model selects documents from a database in response to a user's query and ranks these documents according to their correlation with the query. Query formulation is a significant factor in the successful search of a document collection. Relevance feedback, which is based on the automatic reformulation of the query after the user has evaluated a small subset of the returned documents for relevance, is a method known to provide significant improvement in terms of retrieval effectiveness. The purpose of this research is to study the effectiveness of query reformulation using different feedback methods on the TREC (Text Retrieval Conference) collection of some 1,000,000+ documents. All of this work is an attempt to verify the results produced by Singhal [9]. Singhal used the small, standard Smart collection for his experiments. This work uses the TREC collection and attempts to verify that the conclusions reached by Singhal are valid in this very large database environment.

Table of Contents

Chapter 1	Introduction	1
Chapter 2	Related Work	4
	2.1 Introduction to TREC	4
	2.2 Relevance Feedback	7
	2.3 Relevance Feedback Methods	10
	2.4 Relevance Feedback Evaluation	12
	2.5 Results of Previous Work	15
Chapter 3	Relevance Feedback Schemes	19
	3.1 Research Objectives	19
	3.2 The SMART Retrieval System	20
	3.3 Improvement Measurement	24
	3.4 Design of the Experiments	26
	3.4.1 Number of Documents Assessed Vs Performance	26
	3.4.2 Multiple Feedback Iterations	28
	3.4.3 Quality of the Initial Query	29
	3.4.4 Query Revision with and without Relevance Feedback	30

Chapter 4	Results	33
	4.1 Number of Documents Assessed Vs Performance	33
	4.2 Multiple Feedback Iterations	35
	4.3 Quality of the Initial Query	36
	4.4 Query Revision with and without Relevance Feedback	40
Chapter 5	Future Work	45
	5.1 Feedback Methods in Different Iterations	45
	5.2 Weighting Schemes Vs Relevance Feedback	46
Appendix A	Standard Test Collections and Previous Results	47
	Table A-1 Results obtained by Salton and Buckley	47
	Table A-2 Characteristics of the Standard Test Collections	48
Appendix B	Improvements of Various Feedback Methods on Multiple Feedback Iterations	
	B-1 Improvement of Ide-dec-hi Feedback Method	
	on 5-5-5 Multiple feedback Iterations	49
	B-2 Improvement of Ide-regular Feedback Method	
	on 5-5-5 Multiple Feedback Iterations	50
	B-3 Improvement of Rocchio's feedback method	
	on 5-5-5 Multiple Feedback Iterations	51

Appendix C	
C-1 Quality of the Initial Query (On 5 Documents)	52
C-2 Quality of the Initial Query (On 10 Documents)	54
C-3 Quality of the Initial Query (On 15 Documents)	56
Appendix D	
D-1 Experiment Environment under Smart Version 13.0	57
D-2 Experiment Environment under Smart Version 11.0	57
References	58

Chapter 1

Introduction

Information retrieval can be viewed as searching a high-dimensional document space to bring relevant document to users. It is known, however, this is not an easy task. Due to the complexity of writing styles and the difficulty users have in presenting their information requests, the retrieval results often frustrate users. Generating a modified query is unavoidable if a user wants to improve the retrieval results. However, most systems provide little or no guidance to the user in modifying the original query. Using relevance feedback to help users solve the problem has long been viewed as a viable approach.

Relevance feedback, started more than twenty years ago, has attracted more interest in recent years. With advances in computer hardware and software, the feedback process, which requires more compute time and a friendlier interface than the traditional method, will dominate the information retrieval area.

However, feedback techniques also need to be improved. They should include more intelligent mechanisms, which adapt to the changing environment and handle the feedback processes automatically.

Information retrieval systems are based on an information retrieval model. One of the major information models is the vector space model [1-4], in which the information contained in each document is represented as a content term vector. A content term refers to a keyword or a phrase or a concept that can be used to identify information relevant to the user's needs. The system compares the query vector with each document vector in the

collection, retrieves the documents, which are similar to the query, and returns these documents to the user in rank order.

Query formulation is a significant factor in a successful search of the document collection. All things being equal, a properly formed query will retrieve more relevant documents from the document collection than a poorly formed query. Most users know little about the composition of a collection which they desire to search, nor do they understand the retrieval model on which a retrieval system is based. Therefore, formulating a good query for retrieval purposes in the initial stages of retrieval is not always possible for a user.

Relevance feedback is a process that utilizes the user's relevance assessments to modify the initial query in order to improve it. In this feedback scheme, the user is permitted to review the documents retrieved by the original query and designate those which are relevant to his/her information needs. Based on these relevance judgements, the system automatically modifies the original query. The modified query is then used for a subsequent search of the database.

This thesis presents a detailed analysis of relevance feedback to determine its usefulness in a non-passive user environment. An introduction to TREC, evaluation of relevance feedback, and various methods of relevance feedback are explained in Chapter 2. Related work that has been done in relevance feedback is also described.

Chapter 3 gives the problems of evaluating the performance of relevance feedback and describes an alternative method of calculating improvement in relevance feedback. The relationship between the number of documents shown to the user and the retrieval performance of the feedback process is described. Moreover, the effect of conducting several

feedback iterations as opposed to a single one is discussed. Different feedback methods are analyzed with respect to the retrieval effectiveness of the original user query.

The results of experiments that were conducted to analyze effectiveness are presented in Chapter 4. Chapter 5 indicates future work that can be done in this area.

Chapter 2

Related work

2.1 Introduction to TREC

TREC (Text REtrieval Conference) [5-8] is designed to encourage research in information retrieval using large data collections. Two types of retrieval are being examined: (1) retrieval using an "ad hoc" query such as a researcher might use in a library environment, and (2) retrieval using a "routing" query such as a profile to filter some incoming document stream. The first Text REtrieval Conference (TREC-1) was held in November 1992. The TREC conference is held every year.

The goals of TREC are:

- . To encourage research in text retrieval based on large scales test collections.
- . To increase communication among industry, academia and government by creating an open forum for 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.
- . 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.

2.1.1 The TREC Test Collection

Like most traditional retrieval collections, there are three distinct parts to this collection - the documents, the questions or topics, and the relevance judgments or "right answers." The documents are distributed on CD-ROMs with about 1 gigabyte of data on each, compressed to fit.

The following shows the actual contents of each of the three CD-ROMs (disks 1, 2, and 3).

Disk 1

- WSJ -- Wall Street Journal (1987, 1988, 1989)
- AP -- AP Newswire (1989)
- ZIFF -- Articles from Computer Select disks (Ziff Davis Publishing)
- FR -- Federal Register (1989)
- DOE -- Short abstracts from DOE publications

Disk 2

- WSJ -- Wall Street Journal (1990, 1991, 1992)
- AP -- AP Newswire (1988)
- ZIFF -- Articles from Computer Select disks
- FR -- Federal Register (1988)

Disk 3

- SJMN -- San Jose Mercury News (1991)
- AP -- AP Newswire (1990)
- ZIFF -- Articles from Computer Select disks
- PAT--U.S. Patents (1993)

Table 1 shows some basic document collection statistics. Although the collection sizes are roughly equivalent in megabytes, there is a range of document lengths across collections, from very short documents (DOE) to very long (FR). Also, the range of document lengths within a collection varies. For example, the documents from the AP are similar in length, but the WSJ, ZIFF and especially the FR documents have much wider range of lengths within their collections.

2.1.2 The Relevance Judgments in TREC

The relevance judgments are of critical importance to a test collection. For each topic it is necessary to compile a list of relevant documents (hopefully as comprehensive a list as possible). All previous TRECs have used the Jones & Van Rijisbergen's pooling method [10] 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.

Subset of collection	WSJ (disks 1,2) SJMN (disk 3)	AP	ZIFF	FR (disks 1, 2) PAT(disk 3)	DOE
Size of Collection (megabytes)					
(disk1)	270	259	245	262	186
(disk2)	247	241	178	211	
(disk3)	290	242	349	245	
Number of records					
(disk1)	98,732	84,678	75,180	25,960	226,087
(disk2)	74,520	79,919	56,920	19,860	
(disk3)	90,257	78,321	161,021	6,711	
Median number of Terms per record					
(disk1)	182	353	181	313	82
(disk2)	218	346	167	315	
(disk3)	279	358	119	2896	
Average number of Terms per record					
(disk1)	329	375	412	1017	89
(disk2)	377	370	394	1073	
(disk3)	337	379	263	3543	

Table 1: Document statistics

This sample 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.

2.1.3. Evaluation in TREC

An important element of TREC is to provide a common evaluation forum. Standard recall/precision and recall/fallout evaluation methods are used in TREC. A detailed explanation of the measures is included in the appendix of each of the proceedings of the Text REtrieval Conference [5-8].

2.2 Relevance Feedback

Without a detailed knowledge of the characteristics of a collection, it is not easy for a user to formulate a query that is well suited for retrieval purpose. A query whose vector is more similar to the vectors of those documents in the collection which might satisfy the user's information needs (i.e. the relevant documents) will be more effective for retrieval purposes than a query whose vector has low similarity to the vectors of relevant documents.

Although there exist many methods for query modification, the basic idea behind each method is to move the query vector in the vector space closer to the vectors of the documents judged relevant by the user and away from the documents judged non-relevant. Multiple iterations of relevance feedback can be used to refine the query vector.

Relevance feedback [12-17] is used to modify query vectors automatically using both the user's original query and the user's relevance judgements of the documents retrieved by the original query. The user provides the retrieval system with his/her query. The retrieval system considers the initial retrieval as a test run and retrieves the documents most similar to the user's query. These documents are shown to the user and the user judges them for relevance; that is, for each document, the user judges if that document satisfies his/her informational needs. The system uses this knowledge to modify the query vector so that it is

positioned closer to the relevant documents in the document vector space.

Each improved query vector is used for document retrieval in the subsequent retrieval runs. Since such a query vector is a more accurate reflection of the user's needs, it retrieves more relevant documents than the initial query.

To aid in the understanding of the details of this query modification process, consider a mathematical representation of relevance feedback. Given a query vector in the form

$$Q = (q_0, q_1, \dots, q_t)$$

then Q' represents the revised query,

$$Q' = (q_{0'}, q_{1'}, \dots, q_{t'})$$

which has been modified using relevance feedback; q_j' is the modified weight of concept j in the query vector.

It is known [14] that under appropriate assumptions, the optimal query vector (one that retrieves n documents that are relevant to the query out of a total of N documents in a collection) is

$$Q_{opt} = \frac{1}{n} \sum_{\text{Relevant Documents}} \frac{D_j}{|D_j|} - \frac{1}{n} \sum_{\text{Non-Relevant Documents}} \frac{D_j}{|D_j|}$$

where $|D_j|$ is the Euclidian vector length of document vector D_j .

Since the relevance of documents is not known before query formulation, this method is not practical in a retrieval environment. However, a variant of this method can be used to generate the modified query after some documents have been assessed by the user. A feedback query can be formed using the following scheme

$$Q_{Modified} = Q_{Original} + \frac{1}{T} \sum_{\substack{Known \\ Relevant}} \frac{D_j}{|D_j|} - \frac{1}{N-T} \sum_{\substack{Known \\ Non-Relevant}} \frac{D_j}{|D_j|}$$

In this equation, N documents are shown to the user, and T of the N are assessed to be relevant. This is approximation of the optimal query formulation, obtained by adding the vector elements of the documents identified as relevant by the user and subtracting the vector elements of the documents identified as non-relevant.

The basic intent of query modification using relevance feedback is to move the query towards the area in the vector space where most of the documents relevant to that query are located. This movement is accomplished by modifying the term weights in the query vector. The term weights are modified in such a way that the weights of the terms in the non-relevant documents are decreased, and terms present in the documents judged relevant by the user get higher weights. Thus, the terms found in relevant documents are emphasized in the original query, while the terms appearing in the non-relevant documents are de-emphasized.

h

general, for i iterations, the query modifications is represented as

$$Q_{i+1} = Q_i + \beta \times \sum_{\text{Relevance}} \frac{D_j}{|D_j|} - \gamma \times \sum_{\text{Non-relevance}} \frac{D_j}{|D_j|}$$

where β and γ are weight reduction factors to reduce vector weights before addition to the query. In query modification, the relevant documents should play a more important role than the non-relevant documents, so we usually give higher value to β than to γ . Typical choice for β and γ are 0.75 and 0.25, respectively.

2.3 Relevance Feedback Methods

Three traditional methods of modifying a query, which use the vector adjustment techniques discussed in Section 2.2, have been studied in the past. Ide's approaches use the unnormalized form of the basic feedback process proposed by Rocchio [14]. Rocchio's approach uses normalized vector weights for query modification. The Ide-regular approach uses terms from all the non-relevant documents retrieved by the initial query for query modification, while the Ide-dec-hi approach only uses terms from the top-ranked non-relevant document. The use of only the top ranked non-relevant document for query modification in the Ide-dec-hi approach provides a fixed point in the document space away from which the query vector is to be moved. These methods are summarized in Table 2.3-1.

There are two variations of each feedback methods which result in a total of six query modification procedures using relevance feedback. The two variations are a result of the two

means of extending the query. In the "expansion by all terms" approach, all the terms that

Feedback

Method	Description
Ide-dec-hi	Weights of those terms appearing in the relevant documents are added to the query while the weights of those terms appearing in the top-ranked non-relevant document are subtracted from the query.
	$Q_{Modified} = Q_{original} + \sum_{\substack{\text{all} \\ \text{relevant}}} D_j - \sum_{\substack{\text{top-ranked} \\ \text{non-relevant}}} D_j$
Ide-regular	Weights of those terms appearing in the relevant documents are added to the query while the weights of those appearing in non-relevant documents are subtracted from the query.
	$Q_{Modified} = Q_{original} + \sum_{\text{all relevant}} D_j - \sum_{\text{all non-relevant}} D_j$
Rocchio	Reduced weights of those terms appearing in the documents being used for feedback are added to or subtracted from the original query. Weight reduction factors β and γ are given a value between 0 and 1 so that $\alpha + \beta + \gamma = 1.0$ and typically $\alpha = 1.0$, $\beta = 0.75$ and $\gamma = 0.25$.
	$Q_{Modified} = \alpha \times Q_{original} + \beta \times \sum_{\substack{n1 \\ \text{relevant}}} \frac{D_j}{n1} - \gamma \times \sum_{\substack{n2 \\ \text{non-relevant}}} \frac{D_j}{n2}$

Table 2.3-1 Relevance Feedback Methods

occur in the document judged relevant by the user are added to the original query; this approach tends to increase the length of the query vector considerably. It significantly increases the time for later retrieval. An alternatively approach, called "expansion by most common terms," reduces the modified query length by selectively adding terms that occur in the relevant documents. Term selection is based on the occurrence frequency of the terms in the documents judged relevant by the user. A minimum threshold value for this occurrence frequency is chosen, for example, 75 for word concept type expansion and 15 for phrase concept type expansion (Note: Smart 11.0 does not allow for phrase concept expansion). If the occurrence frequency of a term in a document is less than this minimum threshold, it is not added to the updated query. This approach tends to put a check on the length of the modified query, effectively resulting in a faster retrieval run. Term weights can also be used instead of occurrence frequencies as a criterion for term selection. In general, expansion by all terms takes 2 to 10 times longer than expansion by most common term.

2.4 Relevance Feedback Evaluation

To evaluate performance of relevance feedback, the retrieval effectiveness of the feedback query is compared with the retrieval effectiveness of the original query. Two measures used to express retrieval effectiveness are recall and precision [4]. Recall is the percentage of relevant documents retrieved and precision is the percentage of retrieved documents which are relevant. High recall implies that most relevant documents have been retrieved from the document collection, whereas high precision signifies that most of the documents retrieved are relevant. The objective of a retrieval system is to maximize the number of relevant documents retrieved by a query, while minimizing the number of non-

relevant documents retrieved.

$$\text{Recall} = \frac{\text{Number of relevant items retrieved}}{\text{Total number of relevant items in collection}}$$

$$\text{Precision} = \frac{\text{Number of relevant items retrieved}}{\text{Total number of items retrieved}}$$

High values for both recall and precision are desired for each retrieval run. In a system which uses a low threshold value for similarity, most relevant documents are retrieved, resulting in high recall. However, many non-relevant documents are also retrieved, which lowers precision. Conversely, in a system which uses a very high threshold for similarity between the query and the retrieved documents, the precision is high since only a few documents which are very similar to the query are retrieved. But in such a system, the recall value is low since most of the relevant documents are not retrieved from the collection. Therefore simultaneously obtaining both high recall and precision is not possible. So a retrieval system generally aims at optimizing both recall and precision to some acceptable level.

One problem that arises when recall and precision measures are used to evaluate relevance feedback is the so-called artificial ranking effect. Documents that are judged relevant by user after the initial retrieval run are bound to be retrieved again by the modified query, but this time with a much higher ranking. This should not be considered as

improvement attributable to relevance feedback, since the original query was modified in a way that moves it closer to these particular documents in the vector space.

Moreover, to calculate the improvement of relevance feedback in terms of percentages, the three-point-average measurement is used. In this measurement technique, the precision for a retrieval run is measured at three recall points in the retrieval processes: 0.25 (low recall), 0.50 (medium recall) and 0.75 (high recall). These three precision values are averaged to obtain the precision value for the retrieval run. To compare two retrieval runs, the precision values are compared and percentage improvement is calculated. When the three-point-average measurement is used for performance analysis, the entire document collection is ranked against the query (using relevant assessments that exist for standard test collections). So this measurement reflects the ranking of all documents in the collection for a particular query. In Chapter 3, we use a different measurement in our research.

Various solutions to this evaluation problem are available [18]. One solution uses only the residual collection to analyze the feedback results. In this analysis, once the documents retrieved by the initial query are shown to the user, they are no longer considered for retrieval in the feedback retrieval run. The resultant effect is the same as these documents were removed from the collection for the feedback retrieval. In other words, only the residual collection is available for retrieval with the modified query. To evaluate performance of the feedback method, the residual collection is used for retrieval with the modified query as well as the original query, and these results are compared to evaluate the true improvement gained by relevance feedback.

2.5 Results of Previous Work

Earlier experiments in relevance feedback were conducted by Salton and Buckley [17]. In their experiments fifteen documents were used to modify the original query. All experiments were conducted on the SMART information retrieval system using five different standard test collections. Salton and Buckley used the three-point-average with residual collection analysis to calculate the improvement gained by relevance feedback. Results of these experiments are shown in Appendix A Table A-1. The characteristics of these collections are summarized in Appendix A Table A-2.

In [17], Salton and Buckley conclude that query expansion by all terms is generally preferable to query expansion by most common terms in most cases. But since the difference in the performance of these two methods is moderate, the latter should be used when storage space and retrieval time are major factors. The Ide-dec-hi method performs better on the whole, followed by the Rocchio modification and Ide-regular technique.

Salton and Buckley further concluded that collections with smaller average query lengths are more suited for relevance feedback, since the feedback process involves addition of terms from the relevant documents to the query. Also, queries which perform relatively poorly in the initial retrieval operation have more potential for improvement. Collections which have precisely formulated queries yield better feedback results because the set of relevant documents for a query may be concentrated in a small area of the document space.

From Salton and Buckley 's results, one can conclude that relevance feedback is a very important retrieval tool. Significant improvements in the range of 19% to 160% are

obtained.

Singhal studied the effectiveness of relevance feedback from the user's point of view [9]. First he conducted series of experiments to study the effect of the number of documents on relevance feedback performance. These experiments were conducted on five standard test collections available through the SMART system. The average improvements gained by individual feedback methods are listed in Table 2.5-1, and the average improvements obtained for all the collection are shown in Table 2.5-2

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocchio's	Rocchio's
Expand By	all terms	most common terms	all terms	most common terms	all terms	most common terms
5 Docs	12.07%	11.04%	9.97%	7.86 %	12.80%	10.97%
10 Docs	15.97%	13.14%	11.11%	7.43 %	14.80%	11.97%
15 Docs	18.29%	15.84%	12.91%	10.11%	13.89%	12.96%

Table 2.5-1 Average Improvements Obtained by Individual Feedback Methods (Average taken across five document collections)

Feedback based on	CACM	CISI	CRAN	INSPEC	MED	Average
5 Docs	12.01%	12.26%	15.31%	2.42%	11.38%	10.68%
10 Docs	10.56%	8.97%	16.10%	7.18%	18.75%	12.31%
15 Docs	13.91%	13.46%	14.28%	9.01%	19.32%	14.00%

Table 2.5-2 Average Improvements Obtained by Individual Collection

(Average taken across six feedback methods)

Table 2.5-1 indicates that under all circumstances query expansion by all terms yields better results than query expansion by most common terms, so query expansion by all terms should be used whenever possible. Also, the Ide-regular method does not perform as well as the other two approaches studied, so it should not be used when other feedback methods are available. The Ide-dec-hi approach performs better than Rocchio's approach, so it should be chosen over the latter. From Table 2.5-1, Singhal concluded that in all situations, the use of relevance feedback is an effective methods of improving retrieval.

As can be observed from table 2.5-2, relevance feedback attains an improvement of 10.68% when five documents are assessed for relevance. If ten and fifteen documents are assessed, the improvements attained are 12.31% and 14%, respectively. These results indicate that if a user is willing to evaluate a limited number of documents, say ten, it is always beneficial to perform relevance feedback; substantial improvement is achieved irrespective of the number of documents on which feedback is based.

Singhal also conducted experiments to explore the effect of breaking of a single feedback run into multiple feedback iterations, specifically, into two and three iterations. This results are shown in Table 2.5-3 and Table 2.5-4.

Results from Table 2.5-3 indicates that Rocchio's approach benefits the most from the use of multiple feedback iterations but the overall performance of the Ide-dec-hi approach with multiple feedback iterations is better than the other two approaches. The Ide-regular technique has a lower performance than the other two approaches, but its results are

improved through the use of multiple feedback iterations. Singhal concluded that multiple feedback iterations should be used instead of a single feedback run, irrespective of the feedback technique being used.

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocchio's	Rocchio's
Expand By	all terms	most common terms	all terms	Most common terms	all terms	most common terms
15 Docs	18.29%	15.84%	12.91%	10.11%	13.89%	12.96%
5-5-5 Docs	19.08%	17.38%	14.00%	10.66%	20.43%	18.68%
5-10 Docs	20.10%	17.00%	13.09%	10.52%	18.55%	16.35%
10-5 Docs	20.73%	18.46%	14.61%	11.04%	19.39%	17.63%

Table 2.5-3 Average Improvements Obtained by Individual Feedback Methods
(Average taken across five document collections)

Feedback on Collection	CACM	CISI	CRAN	INSPEC	MED	Average
15 Docs	13.91%	13.46%	14.28%	9.01%	19.32%	14.00%
5-5-5 Docs	16.67%	17.53%	17.16%	7.73%	24.45%	16.71%
5-10 Docs	16.45%	15.31%	17.45%	7.01%	23.46%	15.94%
10-5 Docs	18.53%	16.80%	16.02%	9.94%	23.58%	16.97%

Table 2.5-4 Average Improvements Obtained by Individual Collection
(Average taken across six feedback methods)

Chapter 3

Relevance Feedback Schemes

This chapter analyzes relevance feedback schemes from the user's viewpoint. It describes experiments designed to ascertain the effectiveness of different relevance feedback methods in an interactive system environment.

3.1 Research Objectives

The three-point-average measurement uses ranking of the entire document collection to calculate precision. When this measure is used to calculate the improvement of relevance feedback, ranks of all documents contribute towards this improvement for a query. A user normally reviews only a few top-ranked documents retrieved by the system. Therefore, using the entire document collection to calculate the improvement of relevance feedback is not justified from a user's point of view. We first present an alternate scheme [9] for calculating the improvement achieved by relevance feedback.

In an interactive environment, where relevance feedback is used, the time and effort expended by a user in the retrieval process is significant. Analyzing any document consumes both time and effort on the user's part, so it is always desirable that the user be given the least possible number of documents to judge for relevance. One objective of this work is to ascertain if relevance feedback is effective when a low number of documents are shown to the user for feedback purposes.

One goal is to minimize the time spent by a user in the retrieval process; at the same time the other goal of a relevance feedback system is to maximize improvement by query modification. We want to study techniques that can be used to enhance the improvement gained by relevance feedback, without increasing the user's time. If a single feedback run is broken into several feedback iterations such that the total number of documents shown to the user in various stages is same as in the single feedback run, it can be assumed that the user time consumed in the process does not change. Under this assumption, this study aims at determining the effect of multiple feedback iterations on the performance of relevance feedback.

There are various feedback techniques available for query modification. Performance of different feedback techniques must differ in various feedback situations. The objective of the last part of this study is to establish how the individual feedback methods work in different feedback situations. The results of Singal's [9] study suggest a feedback method which is guided by the quality of the initial query.

3.2 The Smart Retrieval System

For over 30 years, the Smart project at Cornell University has been interested in the analysis, search, and retrieval of heterogeneous text databases, where the vocabulary is allowed to vary widely, and the subject matter is unrestricted. Such databases may include newspaper articles, newswire dispatches, textbooks, dictionaries, encyclopedias, manuals, magazine articles, and so on.

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:

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

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.

In choosing a term weight system, low weights should be assigned to high-frequency terms that occur in many documents of a collection, and high weights to terms that are important in particular documents but unimportant in the remainder of the collection. The weight of terms that occur rarely in a collection is relatively unimportant, because such terms contribute little to the needed similarity computation between different texts.

A well-known term weight system following that prescription assigns weight w_{ik} to term T_k in query Q_i in proportion to the frequency of occurrence of the term in Q_i and in inverse proportion to the number of documents to which the term is assigned [20]. Such a weight system is known as a $tf \times idf$ (term frequency times inverse document frequency) weight system. In practice the query lengths, and hence the number of non-zero term weights assigned to a query, vary widely.

The terms T_k included in a given vector can in principle represent any entities

assigned to a document for content identification. In the Smart context, such terms are derived by a text transformation of the following kind:

1. Recognize individual text words
2. Use a stop list to eliminate unwanted function words
3. Perform suffix removal to generate word stems
4. Optionally use term grouping methods based on statistical word co-occurrence or word adjacency computations to form term phrases (alternatively syntactic analysis computations can be used)
5. Assign term weights to all remaining word stems and/or phrase stems to form the term vector for all information items.

Once term vectors are available for all information items, all subsequent processing is based on term vector manipulations.

The fact that the indexing of both documents and queries is completely automatic means that the results obtained are reasonably collection independent and should be valid across a wide range of collections. No human expertise in the subject matter is required for either the initial collection creation or the actual query formulation.

Text Similarity Computation

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

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

Thus, the similarity between two texts (whether query or document) depends on the weights

of coinciding terms in the two vectors.

To construct this research, the Smart retrieval system and the TREC collection are used. The Smart retrieval system was initially designed at Harvard University in 1961. In 1965 the project was moved to Cornell University. The full system became operational on an IBM 7094 computer in 1964. In the early 1970s, a version of Smart was designed for IBM 360 and 370 machines. A partial UNIX-based version was later developed for DEC's PDP 11-80 and VAX 780 machines. A version of Smart is implemented on the UNIX platform on various VAX machines. Most recently, Smart has been implemented on the Sun workstations. With time, additional information models (e.g., the extended Boolean model) were added to Smart. Various advances in information retrieval have also been added to Smart. On the whole, Smart provides a flexible environment for research in information retrieval.

The experiments conducted for this study use the features of Smart that are associated with the vector space model. The TREC collection is present in the system in the form of term frequency weighted vectors. The TREC collection has a set of query vectors associated with it, which are also weighted by term frequency. Relevance assessments for these queries are done by Jones & Van Rijisbergen's spooling method [10]. These term frequency weighted documents and queries have been re-weighted using the following, improved weighting scheme which uses the inverse document frequency of the concept terms [9]:

$$W_{ij} = \frac{0.5 + 0.5 \frac{tf_{ij}}{(\max tf)_j} \cdot \log \frac{N}{df_j}}{\sqrt{\sum_{k=1} (0.5 + 0.5 \frac{tf_{kj}}{(\max tf)_j})^2 \cdot \log \frac{N}{df_k}}}$$

where W_{ij} is the weight of term i in document j , tf_{ij} is the frequency of the term i in the document j , $(\max tf)_j$ is the maximum occurrence frequency of any term in document j , N is the number of documents in the collection, and df_j is the document frequency of term i in the document collection.

Recall that the term frequency weight scheme does not differentiate between terms that appear in many documents with a high frequency and those appear in certain documents with low frequency. Introduction of inverse document frequency to the weight reduces the weight for such terms. For a retrieval run, similarities are computed between each query and the entire document collection. The inner-product similarity measure (described earlier) is used for this computation. For each query, documents of the entire collection are ranked in decreasing order of their similarity to the query. The user specifies the number of documents that he/she wants to be displayed by the system, and that number of documents is retrieved for the query.

3.3. Improvement Measurement

The improvements in relevance feedback obtained by Salton and Buckley[17] are based on the three-point-average measurement. Recall that the three-point-average is calculated by averaging the precision of a retrieval run at three different recall points (say,

0.25, 0.50 and 0.75). This requires a performance ranking all the documents in the collection for a given query. By this method, if a document is ranked 80 in order of similarity to the initial query, and if after query modification the same document is ranked 52nd, the improvement in rank will contribute to the improvement achieved by relevance feedback. This means of calculating the improvement of relevance feedback is acceptable if the process is being analyzed from a theoretical point of view only.

The three-point-average is not appropriate if the user's point of view is considered; seldom will a user elect to analyze such a large number of document (such as 52 in the above example) to find another relevant document for a query. From a user's point of view, the number of relevant documents contained in the total number of documents displayed is more important. The ranking of the relevance feedback from a user's point of view should be based only on the documents that the user actually sees, rather than the documents that remain undiscovered in the document collection and are never shown to the user. Under this assumption, the improvement gained by relevance feedback can be calculated by considering the difference in the number of relevant documents that the user sees without relevance feedback and the number presented with relevance feedback. This is the approach taken in Singhal's thesis [9].

All the improvements quoted in this study are based on this principle. For example, suppose relevance feedback is performed using the relevance judgements for the ten top-ranked documents initially retrieved. Suppose the user judges five documents out of these ten to be relevant. The initial query is then modified using these ten documents and their relevance judgements. Suppose further that another ten documents are retrieved from

the residual collection (using this modified query) and that five of those ten are relevant. How do we calculate the improvement gained by the modified query? We compare the performance of the modified query with that of the initial query [9]. Suppose the retrieval of another ten documents on the residual collection with the initial query yields three relevant documents. So without using relevance feedback, the user gets a total of eight relevant documents out of the twenty documents retrieved, while if relevance feedback is used, the user gets a total of ten relevant documents. So from the user's point of view, the improvement gained by relevance feedback is

$$(10-8)/8 = 25.0\%$$

This improvement measure is used throughout this study.

3.4 Design of the Experiments

This section describes the design of the experiments used in this study. Section 3.4.1 describes the experiments used to study the effect of the number of documents shown to the user in various feedback processes. Section 3.4.2 discusses multiple feedback iterations. Section 3.4.3 is aimed at studying the effect of the quality of the initial user query on the process of relevance feedback. Section 3.4.4 describes the set of experiments that use relevance feedback, top ranked documents and half-top-ranked documents with query reformulation.

3.4.1 Number of Documents Assessed Vs Performance

As discussed earlier, it is preferable that the user be given the least possible number of documents to judge for relevance. But if too few documents are used for query

modification, the improvements achieved by relevance feedback might not be significant. Therefore, it is desirable for the user to see a minimum number of documents while still achieving improvement through relevance feedback process. The improvement gained by relevance feedback does not necessarily increase linearly with number of documents shown to the user, which are also used for query modification. So if the number of documents on which the feedback is based is increased, there exists a range within which significant improvement is gained by relevance feedback. If the number of documents is further increased beyond this range, the improvement of relevance feedback does not improve in the same proportion.

In Section 3.4.2, we analyze the performance of relevance feedback when five, ten or fifteen documents are initially shown to the user for assessment. Each set of experiments consists of performing searches for all the queries in the TREC document collections, using the six feedback methods. This represents a total of 630 query modifications.

```
Automatically index query collection QC
For each number of documents wanted
Do initial run using the query set QC
  For each query Q in query set QC
    Compute similarity of Q to each of documents in indexed data set
    keep track the top N retrieved documents, sorted by similarities
  For each feedback method
    Generate feedback query (Modified query)
    Perform feedback retrieval on residual collection
  Iterate
```

Figure 3.1 Algorithm for multiple feedback iterations

The results of these experiments are discussed in Section 4.1.

3.4.2 Multiple Feedback Iterations

One advantage of relevance feedback is that the retrieval process can be subdivided into several iterations. To perform feedback in successive iterations, the retrieval process begins with the original query and for each iteration, documents are retrieved and shown to the user; the query obtained from the previous iteration is modified using the relevance judgements of those documents. A modified query is then formed for the next iteration. We want to study the effect of multiple feedback iterations on the performance of relevance feedback.

In this set of experiments, the feedback process, based on the initial retrieval of fifteen documents, is first broken into three iterations, each based on a retrieval of five documents. For example, the initial query (Q_0) is used to retrieve five documents, which are shown to the user. Using relevance assessments of these five documents, Q_0 is modified to yield a new query Q_1 . Q_1 is used to retrieve another five documents which are again shown to the user; relevance assessments are used to modified Q_1 which becomes the new query Q_2 , etc. The final query, Q_3 , is used to retrieve fifteen more documents from the collection. In this way, the user judges fifteen documents for relevance and sees a total of thirty documents. The query modification takes place three times instead of just once. This process is illustrated in Figure 3.2

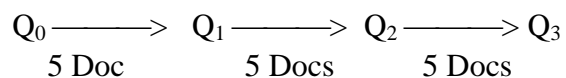


Figure 3.2 Three feedback iterations using five documents each

The following figure shows the typical feedback method in which fifteen documents are displayed and the query is modified.

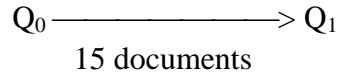


Figure 3.3 Single feedback iteration using fifteen documents

Now the effectiveness of Q_3 in Figure 3.2 is compared with the retrieval effectiveness of Q_1 in Figure 3.3 to find the effect of multiple feedback iterations as compared to a single feedback run. Section 4.2 also analyzes the improvement on different iterations by using different relevance feedback methods.

Notice that in the make feedback query stage, we can use different feedback methods, such as Ide-regular or Rocchio's method, with expansion by all terms or most common terms.

3.4.3 Quality of the Initial Query

The "quality" of a query is a function of the percentage of relevant documents it retrieves. We want to ascertain the relationship between query quality and relevance feedback. We expect to find that some methods perform better for good queries than for poor queries. Therefore the user's relevance judgements of the initially retrieved documents can be used by the system to choose the best possible feedback techniques to modify the initial query.

To perform this experiment, the queries used in our experiments are broken down into various groups. The groups are formed on the basis of the number of relevant documents returned by a query in the initial retrieval of five, ten or fifteen documents. For example, one query group is the set of all the queries that return two relevant documents in the initial retrieval of five documents. Query sets are also formed for the queries that returned none, one, two, three, four, or five relevant documents in the initial retrieval of five,

ten, or fifteen documents. The actual sets formed are given in the next chapter. All the query sets are modified using various feedback methods, and the improvements are analyzed. The results that were obtained from these experiments are presented in Section 4.3.

3.4.4 Query Revision with and without Relevance Feedback

The above sections discussed the performance improvements using relevance feedback. This section uses the information from the top ranked documents to modify the query. This set of experiment consists of three parts. For comparison purposes, the first experiment uses the relevance assessments (Figure 3.1 without iteration). The second (see Figure 3.4) and the third parts use the top-N and the first half of the top-N retrieved documents as a reference for query revision.

To perform the first experiment, first we use the initial query Q_0 to retrieve against the document collection to get, for example, 5 documents. Then we assess the relevance of these 5 retrieved documents. Using these relevance judgements a new query Q_5 is formed. Now we use Q_0 to retrieve 25 documents on the residual collection and use modified query Q_5 to retrieve another 25 documents against the residual collection. The user has seen 30 documents in both cases. Now we can compute the performance improvement using the calculation method discussed in Section 3.3.

Similarly, we use initial query Q_0 to retrieve 10 documents. Based on relevance assessment on these 10 documents, a new query Q_{10} is formed. Now we use Q_0 to retrieve 20 documents on the residual collection and use modified query Q_{10} to retrieve another 20 documents against the residual collection. The user has seen 30 documents in both cases. Again we can compute the performance improvement using the evaluation method discussed

in Section 3.3. Obviously we can compare the performance between the Q_0/Q_5 pair and the Q_0/Q_{10} pair since they all retrieved 30 documents. This idea is summarized below in Table 3.4.4-1.

Initial query	New query formed by using relevance feedback	Remark
Q0	Q5	retrieve next 25
Q0	Q10	retrieve next 20
Q0	Q15	retrieve next 15
Q0	Q20	retrieve next 10
Q0	Q25	retrieve next 5

Table 3.4.4-1 Query revision with relevance feedback

The results obtained in the above experiments are discussed in Chapter 4, Section 4.4.

The second set of experiments is similar to those of in the first set of experiments except that we do not use relevance assessment. Instead we assume the top N ranked documents from the initial query Q_0 are relevant and use this information to form new queries Q_5 , Q_{10} , Q_{15} and Q_{25} , etc. After the new queries are formed, we use them to do retrieval on the residual collection such that the total number of documents shown to the user is always 30. This experiment is summarized below in Table 3.4.4-2.

The third set of experiments are similar to those in the second set of experiments except that we assume the first half of the top N ranked documents from the initial query

Initial query	New query formed by using top N ranked documents	Remark
Q0	Q5	retrieve next 25
Q0	Q10	retrieve next 20
Q0	Q15	retrieve next 15
Q0	Q20	retrieve next 10
Q0	Q25	retrieve next 5

Table 3.4.4-2 Query revision using top N ranked documents

Q₀ are relevant and use this information to form new queries Q₅, Q₁₀, Q₁₅ and Q₂₅, etc. After the new queries are formed, we use them to do retrieval on the residual collection such that the total number of documents shown to the user is 30. The idea is summarized in Table 3.4.4-3.

Initial query	New query formed by using first ½ top N ranked documents	Remark
Q0	Q5	retrieve next 25
Q0	Q10	retrieve next 20
Q0	Q15	retrieve next 15
Q0	Q20	retrieve next 10
Q0	Q25	retrieve next 5

Table 3.4.4-3 Query revision using first ½ top N ranked documents

The results obtained in the above designed experiments are discussed in Chapter 4, Section 4.4.

Chapter 4

Results

The results obtained from running the experiments described in Chapter 3 are described in this chapter. Section 4.1 shows the results of relevance feedback using different numbers of documents. Section 4.2 describes results of breaking a single feedback run into multiple feedback iterations (as showed in Section 3.4.2). Section 4.3 discusses the performance of relevance feedback based on the quality of the initial query (these experiments were described in Section 3.4.3). Section 4.4 presents and discusses the results of experiments designed in Section 3.4.4.

4.1 Number of Documents Assessed Vs Performance

To study the effect of the number of documents on relevance feedback performance, experiments were conducted on TREC test collection using Smart system. The average improvements obtained by individual feedback methods are shown in Table 4.1-1. The experiments were conducted on Smart 13.0 against the TREC collection using queries 91 to 125. The document weight is lnu and the query weight is ltu. The information about the experiment environment is listed in Appendix D-1.

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocc-hio's	Rocc-hio's	Average
Expand By	all terms	most common terms	all terms	most common terms	All terms	Most common terms	
5 Docs	4.17%	-1.04%	5.21%	2.08%	4.17%	2.08%	2.78%
10 Docs	5.45%	4.85%	7.27%	9.09%	6.67%	8.48%	6.97%
15 Docs	4.20%	2.10%	7.56%	7.14%	7.56%	7.98%	6.09%
Average	4.61%	1.97%	6.68%	6.10%	6.13%	6.18%	5.28%

Table 4.1-1 Average Improvements Obtained by Individual Feedback Method
(Average taken across all queries)

From Table 4.1-1, it can be seen that in total 18 experiments, 17 of them benefit from relevance feedback. The only negative gain in the experiments appears in Ide-dec-hi feedback method with 5 documents shown to the user. One can also observe that in most cases, expansion by all terms get higher improvement than expansion by most common terms (6 : 3 in favor of expansion by all terms.)

The improvement ratios in expansion by all terms and expansion by most common terms are very close in the relevance feedback method. The highest average improvement among all the six relevance feedback methods is 6.97% in the 10 document category. The highest average among all the three document categories (5, 10 and 15 documents shown to the user) occurs in Ide-regular with expansion by all terms.

Overall, if the user assesses 10 documents each time, that suffices to get significant improvement by using the relevance feedback method. It is suggested that the user use

expansion by all terms instead of using expansion by most common terms. The results achieved in this set of experiments are close to those of Singhal [9].

4.2 Multiple Feedback Iterations

As described in Section 3.4.2, experiments were conducted on TREC collection in order to study the effect of multiple feedback iterations compared with the performance of single feedback run. The experiments explore the effect of breaking of a single feedback run based on fifteen documents into three successive feedback iterations of five documents each. The experiments were conducted on Smart 13.0 against the TREC collection using queries 91 to 125. The document weight is $\ln u$ and the query weight is $\ln v$. The statistic information about the experiment is listed in Appendix D-1. The average improvements obtained of individual feedback methods are listed in Table 4.2-1.

Results from Table 4.2-1 indicate that performance of relevance feedback is improved by the use of multiple feedback iterations and that multiple feedback iterations should be employed whenever the user elects to assess at least 5 documents for feedback purposes. Under this condition, with no additional effort on the user's part, extra improvement is achieved by relevance feedback.

Moreover, results in Table 4.2-1 indicate that the Ide-dec-hi method benefits the most from the use of multiple feedback iterations followed by the Ide-regular feedback method and Rocchio's method. These results lead one to conclude that multiple feedback iterations should be used instead of a single feedback run, irrespective of the feedback technique being used.

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocc-hio	Rocc-hio	Average
Expand By	all terms	Most common terms	All terms	most common terms	all terms	most common terms	
15 Docs	4.20%	2.10%	7.56%	7.14%	7.56%	7.98%	6.09%
5-5-5 Docs	35.29%	37.39%	30.25%	24.79%	23.95%	25.63%	29.55%
5-5-5 Docs Over 15 Docs	29.84%	34.57%	21.09%	16.47%	15.23%	16.34%	22.26%

Table 4.2-1 Average Improvements Obtained by Individual Feedback Methods
(Average taken across all queries)

The average improvements in table 4.2.-1 are higher than those of Singhal's results (Section 2.5 Table 2.5-3).

4.3 Quality of the Initial Query

To study the retrieval effectiveness of the initial query on the performance of relevance feedback, the queries present within the TREC collection were first divided into various subsets based on the number of relevance documents returned in the initial search of the collection. One group of subsets was formed based on the initial retrieval of five documents, another was based on the initial retrieval of ten documents, yet another group was composed of the subsets formed by the initial retrieval of fifteen documents. Each group

of subsets contained individual queries which returned none, one, two, three, four, or five relevant documents in the initial search. The experiments were conducted on Smart 13.0 against the TREC collection. The document weight is $\ln u$ and the query weight is $\ln t$. The statistics about the experiment are listed in Appendix D-1. The query subsets are listed in Table 4.3-1.

#of queries that retrieve	5 Docs	10 docs	15 Docs	Remark
0 relevant docs.	4	3	2	
1 relevant docs.	4	3	2	
2 relevant docs.	7	3	2	
3 relevant docs.	8	5	3	
4 relevant docs.	6	4	4	
5 relevant docs.	6	4	4	
more than 5 relevance docs	0	13	18	discarded
Total number of queries	35	35	35	

Table 4.3-1 Number of queries in query subsets for TREC collection

Queries from 91 to 125 were used in all the experiments, which makes a total of 35 queries. Let us refer to the above query subsets by $Q_{i/j}$ where the queries of subset $Q_{i/j}$ retrieve i relevant documents in the initial retrieval of j documents. So i varies from 0 to 5 while j can have values 5, 10 and 15.

This query division process resulted in 18 query subsets, i.e. $Q_{0/5} - Q_{5/5}$, $Q_{0/10} - Q_{5/10}$, $Q_{0/15} - Q_{5/15}$. Six experiments were conducted with each of the 18 query subsets (one for each

feedback method), resulting in a total of 108 experiments.

To obtain a correlation between the retrieval effectiveness of the initial query and the feedback methods, the results of experiments conducted on these query subsets are averaged across the individual subsets. The results are shown below.

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocchio	Rocchio	Average
Expand By	all terms	most common terms	all terms	most common terms	all terms	most common terms	
Q _{0/5}	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Q _{1/5}	50.00%	0.00%	50.00%	25.00%	25.0%	25.0%	29.17%
Q _{2/5}	-14.3%	-7.14%	-14.3%	-14.3%	-14.29%	-7.14%	-11.91%
Q _{3/5}	12.50%	8.33%	12.50%	8.33%	12.50%	12.50%	11.11%
Q _{4/5}	4.17%	0.00%	8.33%	4.17%	8.33%	4.17%	4.86%
Q _{5/5}	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 4.3-2 Average Improvements Obtained by Individual Feedback Methods Based on initial retrieval of five documents

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocchio	Rocchio	Average
Expand By	all terms	most common terms	all terms	most common terms	all terms	most common terms	
Q _{0/10}	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Q _{1/10}	00.0%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Q _{2/10}	33.3%	16.67%	16.67%	33.33%	16.67%	33.33%	25.0%
Q _{3/10}	0.00%	6.67%	0.00%	13.33%	6.67%	6.67%	5.56%
Q _{4/10}	-12.5%	-25.0%	-6.25%	0.00%	-6.25%	-18.75%	-11.46%
Q _{5/10}	15.0%	25.0%	15.0%	20.0%	10.0%	10.0%	15.83%

Table 4.3-3 Average Improvements Obtained by Individual Feedback Methods Based on initial retrieval of ten documents

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocchio	Rocchio	Average
Expand By	all terms	most common terms	all terms	most common terms	all terms	most common terms	
Q _{0/15}	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Q _{1/15}	-100%	-100%	0.00%	-100%	0.00%	50.0%	-41.67%
Q _{2/15}	25.0%	75.0%	50.0%	50.0%	25.0%	25.0%	41.67%
Q _{3/15}	0.00%	11.11%	22.22%	22.22%	22.22%	22.22%	16.67%
Q _{4/15}	-37.5%	-43.8%	-25.0%	-25.0%	-37.5%	0.00%	-28.12%
Q _{5/15}	15.0%	-10.0%	15.0%	10.0%	20.0%	20.0%	11.67%

Table 4.3-4 Average Improvements Obtained by Individual Feedback Methods Based on initial retrieval of fifteen documents

It can be observed that if no relevant documents are retrieved in the initial run, the

query cannot be improved using relevance feedback. So to attain some improvement by relevance feedback, there must be some relevant document(s) in the initially retrieved documents. From Tables 4.3-2 to 4.3-4, we observe that if the original query is poorly formulated, it retrieves fewer relevant documents ($Q_{1/n}$ or $Q_{2/n}$) and the feedback method can improve it substantially. Therefore, relevance feedback performs well for poor queries. If the initial query is relatively well formed, it retrieves more relevant documents initially ($Q_{4/n}$ or $Q_{5/n}$) but has less room for improvement since its original retrieval effectiveness is higher. Therefore, for such queries, we obtain smaller improvements.

By comparing the results for the individual feedback methods, we see that in most cases, query expansion by all terms works better than query expansion by most common terms for those query subsets that retrieve fewer relevant documents in the initial retrieval.

The results in this section are not as good as those in Singhal's thesis [9]. The main reason is there are some subsets that have fewer queries than those in Singhal's experiments. For example, we have some subsets that have only two queries while these kind of subsets were excluded in Singhal's experiments.

4.4 Query Revision with and without Relevance Feedback

In Section 3.4.4, we designed experiments to explore the effectiveness of query revision by using relevance feedback, top ranked documents and the first half top ranked documents. The results obtained according to the designs are listed in this section and the observations made from these data are presented. All the experiments discussed in this section were conducted on Smart version 11 against TREC collection using queries 91 to 125. The document weight is atc and the query weight is atc. The statistics about the

experiment is listed in Appendix D-2. The results of this set of experiments are shown in Table 4.4-1, Table 4.4-2 and Table 4.4-3.

Feedback Method	Ide-dec-hi (%)	Ide-dec-hi (%)	Ide-regular (%)	Ide-regular (%)	Rocchio (%)	Rocchio (%)	Average (%)
expand by	All terms	most common terms	All terms	Most common terms	All terms	most common terms	
5-25	-24.42	-26.74	-3.49	-11.63	19.77	12.79	-5.62
10-20	38.37	23.26	56.98	33.72	61.63	32.56	41.09
15-15	67.44	43.02	69.77	46.51	68.60	47.67	57.17
20-10	59.30	44.19	61.63	40.70	62.79	41.86	51.74
25-5	36.05	29.07	37.21	27.91	36.05	26.74	32.17

Table 4.4-1 Query revision using relevance feedback on single iteration
Average over 35 queries

From Table 4.4-1 we observe that query modification using relevance feedback achieved significant improvement in most cases, especially in those middle ranges (that is, the 15-15 and 20-10 cases). The highest improvement appears in the Ide-regular feedback method with initial retrieval of 15 documents and using the modified query to retrieve 15 documents on the residual collection (69.77%). The highest average improvement occurs in the 15-15 cases. In all the cases, expansion by all terms gets better improvements than expansion by most common terms. We also see that only Rocchio's method produces positive improvements on the 5-25 experiments.

Feedback Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regu-lar	Rocc-hio	Rocc-hio	Aver-age
expand by	all terms	most common terms	all terms	most common terms	All terms	most common terms	
5-25	12.79	4.65	11.63	6.98	11.63	6.98	9.11
10-20	18.60	11.63	19.77	10.47	18.60	9.30	14.73
15-15	25.58	11.63	24.42	12.79	22.09	13.95	18.41
20-10	18.60	17.44	17.44	17.44	18.60	18.60	18.02
25-5	9.30	9.30	10.47	9.30	10.47	9.30	9.69

Table 4.4-2 Query revision using top N ranked documents on single iteration
Average over 35 queries

Table 4.4-2 shows the results of experiments using all the top ranked documents to revise the initial queries. We see that expansion by all terms produces better or equivalent improvement compared to the expansion by most common terms. The highest improvement appears in Ide-dec-hi method with expansion by all terms (15-15 case) while the lowest improvement occurs in the case of Ide-dec-hi method with expansion by most common terms (5-25 case). The largest average improvement across six feedback methods was shown in the 15-15 case. We see the lowest improvement occurs in 5-25 case. From Table 4.4-2 we can conclude that if we assume all the top ranked documents are relevant and are willing to use 10 or above initially retrieved documents, we can get some improvement using the six feedback methods.

Feed-back Method	Ide-dec-hi	Ide-dec-hi	Ide-regular	Ide-regular	Rocc-hio	Rocc-hio	Average
expand by	all terms	most common terms	all terms	most common terms	All terms	most common terms	
5-25	-34.88	-37.21	-20.93	-24.42	-11.63	-13.95	-23.84
10-20	0.00	-6.98	0.00	-2.33	9.30	3.49	0.58
15-15	25.58	9.30	23.26	9.30	24.42	10.47	17.05
20-10	19.77	17.44	20.93	16.28	20.93	16.28	18.60
25-5	11.63	11.63	13.95	9.30	13.95	6.98	11.24

Table 4.4-3 Query revision using first half of top N ranked documents on single iteration
Average over 35 queries

Table 4.4-3 shows the results of experiments using the first half of the top ranked documents to revise the initial queries. We see that expansion by all terms produces better or equivalent improvement as the expansion by most common terms. The highest improvement appears in Ide-dec-hi method with expansion by all terms (15-15 case) while the lowest improvement occurs in the case of Ide-dec-hi method with expansion by most common terms (5-25 case). The highest average improvement across six feedback methods was shown in the 10-10 case. We see the lowest improvement occurs in the 5-25 case. From Table 4.4-3 we can conclude that if we assume the first half of the top ranked documents are relevant and

are willing to use 15 or above initially retrieved documents, we can get some improvements using all the six feedback methods.

Overall, we see that query revision using relevance feedback achieve higher average improvements than query revision using top ranked documents and using the first half of the top ranked documents (except in the 5-25 case, in which case query revision with top ranked documents performs the best). Compared with query modification using the first half of the top ranked documents, query revision using top ranked documents has higher average improvements in all cases except the 25-5 case.

Chapter 5

Future Work

The relevance feedback experiments were described in Chapter 3 and the results obtained from running the experiments along with the observations made from the results were presented in Chapter 4. This chapter describes some future work on the TREC collection. Section 5.1 presents a scheme that uses different relevance feedback methods on different feedback iterations. Section 5.2 describes a method to study the impact of different document and query weight schemes on the performance of various relevance feedback methods.

5.1 Feedback Methods in Multiple Iterations

As stated earlier, we can use the number of relevant documents retrieved by a query as a measurement of the quality of that query. We want to ascertain the relationship between query quality and relevance feedback methods in different iterations.

Some methods may perform better for good queries than for poor queries. Therefore the user's relevance judgements of the initially retrieved documents can be used by the system to choose the best possible feedback methods in different stages.

When constructing multiple level feedback queries, we can use different feedback methods, such as Ide-regular, Rocchio's method with expansion by all terms or most common terms. For example, the user could use the Ide-regular in the first feedback iteration. After assessing the retrieved result, the user might choose Ide-regular again in the

second feedback iteration. After making assessments for the newly retrieved documents the user might decide to use Rocchio's method in the third feedback iteration.

We could then analyze the results obtained by the above experiments. We expect that for some queries we can achieve greater improvement by using different feedback methods in the multiple feedback iterations instead of using a single feedback method in the whole feedback process.

5.2 Weight Schemes Vs Relevance Feedback

In the iterative relevance feedback experiments conducted in Chapter 3 the document weight is l_{nu} and the query weight is l_{tu} while for the experiments conducted in Section 3.4.4 both the document weight and query weight are atc . The results that obtained under those conditions were presented in Chapter 4. Different weighting schemes affect the number of relevant documents initially retrieved, so the weighting schemes probably affect the improvement obtained by various relevance feedback methods. To study the relationships between different document/query schemes and relevance feedback methods, we could modify the experiments discussed in Chapter 3 so that different weighting schemes are used. It could be interesting to observe their effect on retrieval.

Appendix A Standard Test Collections and Previous Results

	CACM	CISI	CRAN	INSPEC	MED	Average
No. of Documents	3204	1460	1397	12684	1033	
No. of Queries	64	112	225	84	30	
Initial Run	.1459	.1184	.1156	.1368	.3346	
Ide-dec-hi	.2704	.1742	.3011	.2140	.6350	
all terms	86%	47%	160%	56%	88%	+87%
Ide-dec-hi	.2479	.1924	.2498	.1976	.6218	
most common terms	70%	63%	116%	44%	86%	+76%
Ide-regular	.2417	.1550	.2508	.1936	.6228	
all terms	66%	31%	117%	42%	86%	+68%
Ide-regular	.2179	.1704	.2217	.1808	.5980	
most common terms	49%	44%	92%	32%	79%	+59%
Rocchio	.2552	.1404	.2955	.1821	.5630	
all terms	75%	19%	156%	33%	70%	+70%
Rocchio	.2491	.1623	.2534	.1861	.5297	
most common terms	71%	37%	119%	36%	55%	64%

Table A-1 Results obtained by Salton and Buckley [17]

Appendix A Standard Test Collections and Previous Results (continued)

Document Collection	Number of Vectors	Average Length of Vectors	Average Frequency of Terms in vectors	Percentage of Terms in Vectors with Frequency 1
CACM Documents	3204	24.52	1.35	80.93
CACM Queries	64	10.80	1.15	88.63
CISI Documents	1460	46.55	1.37	80.27
CISI Queries	112	28.29	1.38	78.36
CRAN Documents	1398	53.13	1.58	69.50
CRAN Queries	225	9.17	1.04	95.69
INSPEC Documents	12684	32.50	1.78	61.06
INSPEC Queries	84	15.63	1.24	83.78
MED Documents	1033	51.60	1.54	72.70
MED Queries	30	10.10	1.12	90.76

Table A-2 Characteristics of the Standard Test Collections
Used in Salton and Buckley [17] and Singhal [9]

Appendix B-1 Improvement of Ide-dec-hi feedback method
on 5-5-5 multiple feedback iterations

Qid	Initial retrieved rel#	Ide-dec-hi Most common terms		Ide-dec-hi All terms	
		rel#	Improvement(%)	rel#	Improvement(%)
91	1	1	0.00	1	0.00
92	0	0	0.00	0	0.00
93	12	14	16.67	13	8.33
94	4	7	75.00	8	100.00
95	14	15	7.14	15	7.14
96	15	15	0.00	15	0.00
97	2	3	50.00	2	0.00
98	9	13	44.44	13	44.44
99	10	12	20.00	11	10.00
100	5	7	40.00	4	-20.00
101	3	5	66.67	5	66.67
102	4	6	50.00	6	50.00
103	4	9	125.00	10	150.00
104	3	8	166.67	9	200.00
105	5	7	40.00	5	0.00
106	4	10	150.00	7	75.00
107	5	12	140.00	13	160.00
108	8	12	50.00	15	87.50
109	15	12	-20.00	10	-33.33
110	12	13	8.33	13	8.33
111	14	15	7.14	15	7.14
112	13	11	-15.38	11	-15.38
113	7	11	57.14	11	57.14
114	1	0	-100.00	2	100.00
115	6	11	83.33	10	66.67
116	2	9	350.00	9	350.00
117	9	13	44.44	11	22.22
118	11	6	-45.45	13	18.18
119	6	12	100.00	11	83.33
120	3	1	-66.67	1	-66.67
121	0	0	0.00	2	0.00
122	7	12	71.43	11	57.14
123	12	15	25.00	15	25.00
124	5	15	200.00	15	200.00
125	7	10	42.86	15	114.29
Average improvements:			35.29%		37.39%

Appendix B-2 Improvement of Ide-regular feedback method
on 5-5-5 multiple feedback iterations

Qid	Initial retrieved rel#	Ide-dec-hi Most common terms		Ide-dec-hi All terms	
		rel#	Improvement(%)	rel#	Improvement(%)
91	1	1	0.00	0	-100.00
92	0	0	0.00	0	0.00
93	12	14	16.67	13	8.33
94	4	5	25.00	4	0.00
95	14	15	7.14	15	7.14
96	15	14	-6.67	15	0.00
97	2	3	50.00	3	50.00
98	9	13	44.44	14	55.56
99	10	12	20.00	11	10.00
100	5	7	40.00	4	-20.00
101	3	5	66.67	5	66.67
102	4	6	50.00	5	25.00
103	4	8	100.00	9	125.00
104	3	8	166.67	5	66.67
105	5	7	40.00	5	0.00
106	4	9	125.00	1	-75.00
107	5	12	140.00	13	160.00
108	8	8	0.00	14	75.00
109	15	12	-20.00	12	-20.00
110	12	13	8.33	13	8.33
111	14	15	7.14	15	7.14
112	13	11	-15.38	11	-15.38
113	7	11	57.14	11	57.14
114	1	0	-100.00	0	-100.00
115	6	8	33.33	8	33.33
116	2	9	350.00	10	400.00
117	9	13	44.44	12	33.33
118	11	6	-45.45	8	-27.27
119	6	11	83.33	7	16.67
120	3	0	-100.00	0	-100.00
121	0	0	0.00	0	0.00
122	7	10	42.86	10	42.86
123	12	15	25.00	15	25.00
124	5	15	200.00	14	180.00
125	7	14	100.00	15	114.29
Average improvements:			30.25%	24.79%	

Appendix B-3 Improvement of Rocchio's feedback method
on 5-5-5 multiple feedback iterations

Qid	Initial retrieved rel#	Ide-dec-hi Most common terms		Ide-dec-hi All terms	
		rel#	Improvement(%)	rel#	Improvement(%)
91	1	1	0.00	1	0.00
92	0	0	0.00	0	0.00
93	12	14	16.67	13	8.33
94	4	3	-25.00	7	75.00
95	14	15	7.14	15	7.14
96	15	14	-6.67	15	0.00
97	2	7	250.00	5	150.00
98	9	13	44.44	14	55.56
99	10	12	20.00	11	10.00
100	5	5	0.00	5	0.00
101	3	6	100.00	6	100.00
102	4	3	-25.00	3	-25.00
103	4	5	25.00	5	25.00
104	3	8	166.67	8	166.67
105	5	8	60.00	7	40.00
106	4	1	-75.00	1	-75.00
107	5	12	140.00	12	140.00
108	8	13	62.50	10	25.00
109	15	12	-20.00	12	-20.00
110	12	11	-8.33	11	-8.33
111	14	15	7.14	15	7.14
112	13	11	-15.38	11	-15.38
113	7	11	57.14	11	57.14
114	1	1	0.00	1	0.00
115	6	10	66.67	9	50.00
116	2	8	300.00	9	350.00
117	9	9	0.00	10	11.11
118	11	6	-45.45	12	9.09
119	6	12	100.00	11	83.33
120	3	0	-100.00	0	-100.00
121	0	0	0.00	0	0.00
122	7	10	42.86	11	57.14
123	12	15	25.00	15	25.00
124	5	15	200.00	14	180.00
125	7	9	28.57	9	28.57
Average improvements:			23.95%	25.63%	

Appendix C-1 Quality of the Initial Query (On 5 Documents)

Qid	Initial	Ide-dec-hi		Ide-regular		Rocchio's	
	Retrieved	all	comm	all	comm	all	comm
92	0	0	0	0	0	0	0
114	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0
121	0	0	0	0	0	0	0
subtotal	0	0	0	0	0	0	0
91	1	0	0	1	0	1	0
97	1	2	2	2	2	1	2
108	1	2	1	2	2	2	2
116	1	2	1	1	1	1	1
subtotal	4	6	4	6	5	5	5
94	2	2	2	2	2	2	2
100	2	2	2	2	2	2	2
101	2	2	2	2	2	2	1
102	2	2	1	2	2	2	2
106	2	1	1	1	1	1	1
115	2	2	2	2	2	2	2
125	2	1	1	1	1	1	1
subtotal	14	12	11	12	12	12	11

Appendix C-1 Quality of the Initial Query (On 5 Documents) (continued)

Qid	Initial	Ide-dec-hi		Ide-regular		Rocchio's	
	Retrieved	all	comm	all	comm	all	comm
103	3	3	3	3	3	3	3
104	3	3	3	3	3	3	3
105	3	3	2	3	3	3	3
107	3	3	3	3	3	3	3
113	3	4	5	4	4	4	5
117	3	4	4	4	4	4	4
118	3	3	3	3	3	3	3
124	3	4	3	4	3	4	3
subtotal	24	27	26	27	26	27	2
98	4	5	5	5	5	5	5
110	4	4	4	4	4	4	4
111	4	5	4	5	4	5	4
112	4	4	4	4	4	4	4
119	4	4	4	4	4	4	4
122	4	3	3	4	4	4	4
subtotal	24	25	24	26	25	26	25
93	5	5	5	5	5	5	5
95	5	5	5	5	5	5	5
96	5	5	5	5	5	5	5
99	5	5	5	5	5	5	5
109	5	5	5	5	5	5	5
123	5	5	5	5	5	5	5
subtotal	30	30	30	30	30	30	30

Appendix C-2 Quality of the Initial Query (On 10 Documents)

Qid	Initial	Ide-dec-hi		Ide-regular		Rocchio's	
	Retrieved	all	comm	all	comm	all	comm
92	0	0	0	0	0	0	0
114	0	0	0	0	0	0	1
121	0	0	0	0	0	0	0
subtotal	0	0	0	0	0	0	1
91	1	1	0	1	0	1	1
116	1	2	3	1	2	1	1
120	1	0	0	1	1	1	1
subtotal	3	3	3	3	3	3	3
97	2	4	4	3	3	3	3
100	2	2	2	2	3	2	3
101	2	2	1	2	2	2	2
subtotal	6	8	7	7	8	7	8
94	3	4	4	4	4	4	3
102	3	2	1	2	2	2	2
103	3	2	3	3	3	3	3
104	3	4	4	4	4	4	4
125	3	3	4	3	4	3	4
subtotal	15	15	16	15	17	16	16

Appendix C-2 Quality of the Initial Query (On 10 Documents) (continued)

Qid	Initial		Ide-dec-hi		Ide-regular		Rocchio's	
	Retrieved		all	comm	all	comm	all	comm
105	4		4	3	4	5	4	5
106	4		1	1	2	2	2	3
108	4		6	6	6	7	6	6
115	4		3	2	3	2	3	3
subtotal	16		14	12	15	16	15	17
107	5		4	3	4	4	4	6
117	5		6	6	7	6	6	5
119	5		5	7	5	7	5	7
124	5		8	9	7	7	7	4
subtotal	20		23	25	23	24	22	22

Appendix C-3 Quality of the Initial Query (On 15 Documents)

Qid	Initial	Ide-dec-hi		Ide-regular		Rocchio's	
	Retrieved	all	comm	all	comm	all	comm
92	0	0	0	0	0	0	0
121	0	0	0	0	0	0	0
subtotal	0	0	0	0	0	0	0
91	1	0	0	1	0	1	1
114	1	0	0	1	0	1	2
subtotal	2	0	0	2	0	2	3
97	2	4	4	4	3	3	3
116	2	1	3	2	3	2	2
subtotal	4	5	7	6	6	5	5
101	3	3	2	3	3	3	4
104	3	5	6	4	5	4	4
120	3	1	2	4	3	4	3
subtotal	9	9	10	11	11	11	11
94	4	4	4	5	5	5	6
102	4	2	1	2	2	2	2
103	4	3	3	3	3	3	5
106	4	1	1	2	2	2	3
subtotal	16	10	9	12	12	10	16
100	5	4	4	5	5	5	6
105	5	5	4	5	5	5	5
107	5	4	2	5	5	6	7
124	5	10	8	8	7	8	6
subtotal	20	23	18	23	22	24	24

Appendix D

D-1 Experiment Environment under Smart Version 13.0

Operating system: Sun OS 5.4 / Sun Solaris 2.0

The TREC collection consists of disk1, disk2 and disk3

Weights: Document weight - lnu

Query weight - ltu

Queries: From query 91 to query 125. Total number of queries is 35

Parameter settings:

	Ide-dec-hi	Ide-regular	Rocchio
α	1.0	1.0	1.0
β	1.0	1.0	0.75
γ	1.0	1.0	0.25

D-2 Experiment Environment under Smart Version 11.0

Operating system: Sun OS 5.4 / Sun Solaris 2.0

The TREC collection consists of disk1, disk2 and disk3

Total number of documents in the test collection is 1,316,592

Weights: Document weight - atc

Query weight - atc

Queries: From query 91 to query 125. Total number of queries is 35.

Parameter settings: Same as Appendix D-1.

References

- [1] G. Salton, A. Wong and C.S. Yang, A Vector Space Model for Automatic Indexing, Communications of the ACM, 18:11, November 1975
- [2] G. Salton, *Automatic Information Organization and Retrieval*, McGraw Hill Book Co., Readings, NY, 1968
- [3] G. Salton , *Automatic Text Processing*, Addison-Wesley Publishing Company, Inc. 1989
- [4] G. Salton , M. McGill , *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, 1983
- [5] D. Harman , Overview of the First Text REtrieval Conference(TREC-1), in D. Harman, Editor, *The First Text REtrieval Conference (TREC-1)* ,National Institute of Standards and Technology Special Publication 500-225, 1993
- [6] D. Harman , Overview of the Second Text REtrieval Conference(TREC-2), in D. Harman, Editor, *The Second Text REtrieval Conference (TREC-2)*, National Institute of Standards and Technology Special Publication 500-225, 1994
- [7] D. Harman , Overview of the Third Text REtrieval Conference(TREC-3), *The Third Text REtrieval Conference (TREC-3)* , National Institute of Standards and Technology Special Publication 500-225, 1995
- [8] D. Harman , Overview of the Fourth Text REtrieval Conference(TREC-4), *The Fourth Text REtrieval Conference (TREC-4)* , National Institute of Standards and Technology Special Publication 500-225, 1996
- [9] A. Singhal, Relevance Feedback as A Tool in Information Retrieval. Master's thesis, University of Minnesota, Duluth, February 1992
- [10] S. Jones and V. Rijisbergen C. , *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
- [11] G. Salton and C. Buckley, Term Weight Approaches in Automatic Text Retrieval, Technical Report 87-881, Department of Computer Science, Cornell University, Ithaca, NY, 1987
- [12] E. Ide, New Experiments in Relevance Feedback, in G. Salton, Editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1971
- [13] E. Ide and G. Salton, Interactive Search Strategies and Dynamic File Organization in Information Retrieval, in G. Salton, Editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1971
- [14] J.J. Rocchio, Jr., Document Retrieval System: Optimization and Evaluation, Ph.D. Dissertation, Harvard University, In *Report ISR-10* to the National Science Foundation, Harvard Computational Laboratory, Cambridge, MA, March 1966
- [15] J.J. Rocchio, Jr., Relevance Feedback in Information Retrieval, in G. Salton, Editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1971

- [16] G. Salton, Relevance Feedback and the Optimization of Retrieval Effectiveness, in G. Salton, Editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1971
- [17] G. Salton and C. Buckley, Improving Retrieval Performance by Relevance Feedback. Technical Report 88-898, Department of Computer Science, Cornell University, Ithaca, NY, February 1988
- [18] Y.K Chang, C. Cirillo and J. RaZon, Evaluation of Feedback Retrieval Using Modified Freezing, Residual Collection, and Test and Control Group, in G. Salton, Editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1971
- [19] C. Crouch , D. Crouch and K. Nareddy, The Automatic Generation of Extended Queries, *In Proceedings of the Thirteenth Annual International ACM SIGIR Conference*, Gernoble, France, 1990
- [20] W.B. Croft, J. Callan, and J. Broglio, TREC-2 routing and ad-hoc retrieval evaluation using the INQUERY system. In D. Harman, editor, *The Second Text REtrieval Conference(TREC-2)*. National of Standards and Technology Special Publication 500-225, 1994