INTERACTIVE REFORMULATION OF LONG QUERIES

A Dissertation Presented

by

GIRIDHAR KUMARAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2008

Computer Science

INTERACTIVE REFORMULATION OF LONG QUERIES

A Dissertation Presented

by

GIRIDHAR KUMARAN

Approved as to style and content by:

James Allan, Chair

W. Bruce Croft, Member

David Jensen, Member

Andrew Cohen, Member

Andrew G. Barto, Department Chair
Computer Science

To my parents

ACKNOWLEDGMENTS

This thesis would have been impossible without the guidance, direction, and encouragement of a large number of people including academic advisors, colleagues, relatives, and friends. A complete list would at best be incomplete: I will undertake this acknowledgment knowing that I will miss mentioning a number of people.

I joined the Computer Science Department at University of Massachusetts Amherst with no background in Computer Science, and a vague idea that I wanted to study Artificial Intelligence. Some well-meaning advice from Prof. Shlomo Zilberstien prompted me look around before committing to an area of study, and a course on Web Search Engines taught by Prof. James Allan piqued my interest. I will begin by first thanking my advisor Prof. James Allan for introducing me to the field of Information Retrieval, and agreeing to accept me as his advisee. I am grateful to him for his commitment, the freedom he gave me to pursue my ideas, the encouragement he provided when I succeeded, the patience he demonstrated when I failed, the wide range of problems he exposed me to, and the direction he consistently provided. The extra effort he put into improving my technical writing and honing my presentation skills will always be appreciated. I am indebted to him for providing an atmosphere where I could learn and grow.

Prof. Bruce Croft inspired my experimental methodology and made me aware of the importance of understanding and relating to past work. His presence during my lab presentations was always looked forward to for the insightful observations and comments he made. Prof. David Jensen's ideas on conducting successful research helped accelerate my progress by helping me focus and plan my research carefully.

I would also like to thank Prof. Andrew Cohen for the enthusiasm and interest he showed in my work.

Outside of my department I would like to thank my internship mentors, Dr. Rosie Jones at Yahoo! Research, and Dr. Krysta Svore and Dr. Chris Burges at Microsoft Research. I am grateful to Rosie for providing me an opportunity to work on an interesting problem that introduced me to the utility and power of machine learning. Her infectious enthusiasm in continuing to collaborate after the internship and the ability to apply different approaches to problems had a deep influence on me. Krysta was a wonderful mentor who provided me the challenge and opportunity to work in an amazing research lab on some very interesting problems. Her excellent managerial style revealed aspects to research that I hadn't known before.

The support provided by the Computer Science department and the Center for Intelligent Information Retrieval was fantastic. The efficiency of Graduate Secretaries Sharon Mallory and Leeanne Leclerc never ceased to amaze me. Special thanks to Kate Moruzzi and Jean Joyce for their never-ending patience entertaining requests for appointments, reimbursements, equipment, and poster printing - all at short notice. Andre Gauthier was more of a friend than the Lab Manager who unfailing heeded to my requests for computing support. David Fisher deserves special mention for helping me with Lemur and Indri, and the time he took off to attend dry-runs of my presentations and provide sound advice.

I should also thank my colleagues Mark Smucker, Niranjan Balasubramanian , Ben Carterette, Ao Feng, Yun Zhou, Trevor Strohman, Xiaoyong Liu, Ron Bekkerman, Don Metzler, Hema Raghavan, Fernando Diaz, Jiwoon Jeon, Ramesh Nallapati, and Vanessa Murdock for the academic and personal companionship they provided over the years. My friends in the Indian community with whom I shared many memorable experiences are too numerous to be mentioned individually.

The place of pride in my acknowledgments goes to two wonderful people - my parents Mr. V. Kumaran and Mrs. Shyamala Kumaran . My father was a driving force for my PhD aspirations, and the values of honesty and hard work he instilled in me have always carried me through. My mother, in her own quiet and unassuming way, has been my greatest source of inspiration and strength. Her unflinching faith in me was a constant source of reassurance in times good and bad. The sacrifices they have made for my upbringing and education can never be fully repaid. By dedicating this thesis to them I hope to do a bit in that direction.

Meeting my wife Ramya was the best thing that happened to me in Amherst. I thank her for her patience and courage through these years, and putting her career and interests on hold until I completed my PhD. I must also thank my brothers Gautam and Guhan for steadfastly supporting me, and always daring me to think big.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, in part by NSF Nanotech #DMI-0531171 and in part by an award from Google. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

ABSTRACT

INTERACTIVE REFORMULATION OF LONG QUERIES

MAY 2008

GIRIDHAR KUMARAN

B.E., UNIVERSITY OF MADRAS

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

DOCTOR OF PHILOSOPHY, UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

We present new ways of interacting with a user based on query analysis and reformulation. Our goal is to not only improve retrieval performance but also help the user understand the retrieval process and collection she is searching. We do this by providing users information reflecting the potential impact their decisions will have on the retrieval process. This way, users can make more informed choices from the options presented to them by the retrieval system.

Unlike most previous work in user interaction where a one-procedure-fits-all strategy was pursued, user interaction must be invoked only when there is potential for improvement. This is important as tedious user interaction can have an unfavorable impact on user experience. We present techniques for selective user interaction and show their utility in the context of two interaction techniques we have developed. Our results show that user interaction can be avoided in a vast number of cases without much deterioration in performance.

User interaction can be made more productive by providing users with an optimally-sized set of high quality options. We present efficient techniques to determine such a set. When faced with a decision to interact with a user given a particular query, it is beneficial is determine the best interaction technique suited for that query. We solve this problem by obtaining implicit feedback from the user. By utilizing all the interaction-related techniques described in this thesis, we show through simulations and user studies that users can obtain better performance with less effort.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The dominant paradigm in *ad hoc* information retrieval (IR) is quite straightforward. A user with an information need enters terms in a text box, which the search engine matches with indexed documents guided by some underlying retrieval model. This retrieval paradigm makes a number of assumptions. These include considering the terms the user entered as an accurate description of the information need, believing the terms will guide the search engine towards retrieving *only* relevant content, and presuming that the retrieval model captures human notions of relevance. While decades of research and application of this paradigm have proved its utility and effectiveness, there are still many failure points, and a lot of room for improvement. Some failures can be attributed to the disconnect between what users enter as queries and the ideal representation required to retrieve the documents they want (Nordlie, 1999). The quality of queries submitted to IR systems directly affects the quality of search results generated by these systems (Croft and Thompson, 1987). In conveying complex information needs, users enter queries that would appear perfectly legitimate and understandable to a human being. Unfortunately, in a large number of cases such queries are not handled well by the search engine. While users generally have a model of their information need, they have little or no knowledge about how the underlying search engine works. This lack of knowledge is usually coupled with another unknown - the contents of the collection being searched. All this can lead to search failures as what lies under the hood of a search engine is not an intelligent agent capable of human discourse, but an algorithm that works with the statistics of text. The current paradigm of search, wherein a user enters terms into a text box to describe her information need and the search engine responds by simply returning a list of documents based on some underlying statistical model is prone to failure. This is true not only of specialized environments like medical text, legal information, and enterprise search but also more ubiquitous ones like the web. Irrespective of advances in retrieval models and user interfaces, a critical gap exists between the user's own expression of information need and the best representation suited for retrieving the exact information satisfying that need. Hence there is clearly a need for an unobtrusive intermediary to help bridge the gap between the user and the system.

1.1 Motivation for User Interaction

A large number of automatic methods exist to directly or indirectly fulfill the above-mentioned role. However these have their own particular limitations and are mostly successful in improving retrieval in cases when performance is already satisfactory (Kumaran and Allan, 2006a). This leads us to explore the option of involving the user to a greater extent in the process, hoping that it will lead to more comprehensive improvements in retrieval effectiveness.

There is a continuing and growing case for involving users to a greater degree in the information retrieval process. In a recent study, Harman and Buckley (2004a) investigated the reasons for retrieval failure, and identified a set of ten different common reasons for failure. Given current capabilities, most of these problems can be overcome only by actively involving the user. The diverse nature of these problems also means that there is no single *interaction technique* that can help tackle all these problems. We define an interaction technique as a method for interacting with the user. This includes the logic used to provide options to the user, the supplementary information provided to help the user choose from the options in an implicit or explicit fashion, and the way information gleaned from the user is utilized by the IR system. Given the multiple reasons for retrieval failure, there is need for a suite of interaction techniques to ensure good performance in all environments. Further, the development of this suite of techniques must be done with the user's ability, willingness and cognitive load in mind. Requiring the user to expend unacceptable amounts of participation can be detrimental to the overall user experience.

1.2 Scope of the thesis

This thesis deals with the exploration of interaction techniques for effectively and efficiently bridging the gap between the expression of a user's information need and the search engine they utilize to satisfy that need. The quest is not the design of an ultimate interface or creation of a software agent with artificial intelligence to play the role of an intermediary. It is instead to devise methods to effectively and efficiently involve users in not only translating their information needs into the best representation for input to a search engine but also obtaining feedback on retrieval performance unobtrusively. Our research focuses on the crucial query creation and reformulation step in information retrieval. By way of guiding the formulation of a query we hope to help users issue queries that are best suited for retrieving the information they seek. In addition, such guided query formulation with an appropriate interface can help users understand the collection they are searching and the reasons why the search engine returns the results they see. This way there is hope to involve the user further in the search process, and gradually help them become search experts themselves.

1.3 Queries of Interest

As mentioned in the previous section, we are interested in user interaction for query formulation. In this thesis we focus on user interaction for information retrieval with *long queries*. We define long queries as those with a length greater than three terms, and *short queries* as those with a length less than or equal to three.

Long queries are interesting as they provide a rich expression of a user's information need, and also provide context that can be leveraged to deliver improved search performance. Long queries can be obtained in two ways.

- Users themselves enter long queries.

  While it is well known that the users of web search engines enter queries that are only two to four terms long (Spink et al., 2002), there are a number of domains like health verticals and product help desks where users issue long queries.

- From short queries, when expanded with automatic techniques like local context analysis (Xu and Croft, 2000) and pseudo-relevance feedback (PRF) (Lavrenko and Croft, 2001).

  Query expansion is a well-known technique in information retrieval that involves including additional related terms to an original *short* query to improve retrieval performance. The number of additional terms included can vary from a few to even hundreds, resulting in the short queries being transformed into long ones.

However, handling queries with such a large number of terms is tricky. Users can include terms in the query that end up confusing the search engine, or automatic query expansion techniques can suggest terms that impair effective search. Our fundamental approach to handling retrieval using long queries is to automatically or interactively identify the crux of the query, i.e. a subset of terms from the long query that capture the information need succinctly.

One approach to handling long queries of the first type would be to explore the use of natural language processing (NLP) techniques to parse the query and reformulate it based on grammatical analysis. We instead choose to avoid such complex analysis and treat the terms in the query as independent of each other. As we will show in Chapter 4, this simpler approach yields significant improvements in performance for a wide range of queries.

1.4 Contributions of the thesis

The goal of this thesis is to develop, test and evaluate a suite of very effective interaction techniques that do not demand much effort, cognitive and physical, from the user. In a sense, this involves devising a set of pointed questions to ask the user in response to a query. The accent is on moving away from the traditional feedback paradigm of asking an user to select relevant documents, passages or terms and instead ask for information in a more intelligent fashion. The questions are also meant to be answered in a simple way - usually yes/no decisions conveyed through forms. A recent study (White and Ruthven, 2006) has shown that while users like to have some amount of control over various aspects of the retrieval process, they still prefer unobtrusive ways to communicate during the process.

The main contributions of the thesis are as follows:

1. **Empirical validation of the potential of long queries**. We bring to focus the importance of developing techniques for handling long queries. Commercial search engines, which are optimized for shorter queries, perform worse on long queries compared to short ones. We show that performance can be improved through reformulation of long queries. Such reformulation results in roughly 50% improvement over the baseline, as measured by mean average precision.

2. **User interaction for long queries**. We present a technique based on user interaction to help reformulate long queries. We develop and evaluate a new look-ahead interface designed for long queries. We report the results of user studies using two variants of the reformulation technique, *interactive query reduction* (IQR) and *interactive query expansion* (IQE), that showed the utility of the interface in not only rapidly directing users towards their information needs, but also providing them a preview of the collection contents and a sense of how the retrieval system works.

3. **Selective user interaction**. We show that user interaction in the form of IQR and IQE for every query is not a guaranteed success. When the starting point is weak, i.e. the original query issued by the user is of poor quality, any amount of user interaction is futile. In situations like this, it's appropriate to detect failure in advance and avoid interaction. We present techniques to identify queries for which user interaction is futile, and show that we can avoid interaction for around 40% of queries in our test set without significantly degrading improvements in performance.

4. **Efficient user interaction**. An often-neglected aspect of interactive information retrieval is the background processes needed to analyze and present options to users. We present techniques to convert the exponential-time processes for IQR and IQE to polynomial. This way IQR and IQE can be deployed in operational environments.

5. **Reduced user interaction**. We develop and evaluate techniques to identify and present a minimal set of high-quality options to users during IQR and IQE. We will show that we can obtain the same levels of performance as the baseline by showing 30% and 80% less options for IQR and IQE respectively.

6. **Combining interactive techniques**. We present implicit techniques to determine the better interaction technique between IQR and IQE in response to a query. The effect of doing so is a roughly 50% improvement in performance, with an interface that requires showing users approximately 50% less options than the baseline.

1.5 Organization of the thesis

In Chapter 2 we survey related work in user interaction that pertains to this thesis. We introduce the data sets we work with as well as the evaluation measures we report in this work in Chapter 3. Before describing our automatic techniques to identify the important terms or concepts in a long query in Chapter 5, we provide illustrative examples of the potential for improvement in retrieval performance by using long queries in Chapter 4, and provide empirical validation for the performance improvements that can be realized. The automatic techniques for long query reformulation we explored had limited success. This motivated the investigation of user interaction to overcome the shortcomings of automatic techniques in Chapter 6. In Chapter 7 we present work related to selective user interaction: i.e., techniques to identify the potential utility of user interaction. The interaction techniques IQR and IQE that we present in this thesis require exponential-time background processes. In Chapter 8 we show how we can convert these processes to polynomial, paving the way for their potential use in real-time situations. Choosing the right interaction technique in response to a query is important. In Chapter 9 we show that we can implicitly obtain feedback from the user on the right interaction technique to be invoked, IQR or IQE, in response to a long query. Chapter 10 contains examples of other interaction techniques we explored and evaluated. Conclusions and future work are discussed in Chapter 11.

CHAPTER 2
RELATED WORK

In this chapter we review related work that provides the foundation and motivation for much of this thesis. In Section 2.1 we visit the need for interaction. While automatic techniques for improving information retrieval are quite effective, they are considerably narrow in scope when it comes to the range of queries they can improve performance on. As a prerequisite before venturing to develop interaction techniques, in Section 2.2 we survey past work on understanding user behavior and perceptions. In Section 2.3 we show that there has been a large amount of past work that has successfully made use of user interaction. Sections 2.5 and 2.6 provide an overview of the techniques that previous researchers used to interact with users and obtain useful information from them. Our work differs from work done by previous researchers in the following way.

- We focus on a particular type of queries namely long queries.
- We tackle issues like predicting the utility of user interaction and the efficiency of background processes for IQR and IQE.
- We explore techniques to reduce the amount of interaction keeping the potential for improvement unaffected.
- We show how we can invoke the right interaction technique from a set available in response to a query.

2.1 The need for interaction

A slew of automatic techniques have been developed to improve IR. However, motivating our interest in interactive IR is the Reliable Information Access (RIA) workshop conducted by Harman and Buckley (2004a,b). The goal was to determine the reasons behind the variability in performance across IR systems and queries on a test collection. The workshop revealed that systems utilizing different retrieval models performed non-uniformly on the same queries, and each system showed wide variability across a set of queries. Further analysis showed that each IR system had its strengths and weaknesses, and if the potential problems with a query could be determined in advance, then an appropriate IR system could be used to significantly improve effectiveness. The workshop also identified a number of common reasons for failure (Buckley, 2004), and concluded that certain types of failure were beyond the scope of current IR technology. Listed below are some of the reasons.

- Systems emphasize one aspect; missing another required term.
- Systems emphasize one aspect; missing another aspect.
- Some systems emphasize one aspect; some another; need both.
- All systems emphasize one irrelevant aspect; missing point of topic.
- Need QA query analysis and relationships.
- Systems missed difficult aspect that would need human help.

In the absence of automatic techniques to tackle these types of problems, manual intervention through user interaction is a potential recourse.

With this in mind, we have explored interaction techniques with the goal of improving retrieval performance with minimal participation from the user. We investigated whether asking the user simple questions or requiring the user to make minor modifications to the query will improve performance to an extent that justifies the little additional effort put in by the user. The variability in the types of failures as reported by Harman and Buckley (2004a,b) makes us acknowledge that there might not be one single interaction technique that improves performance across all queries; we believe that a set of techniques will take us closer to that goal.

2.2 User behavior

Nordlie (1999) conducted a study to compare the behaviors of library users interacting with a librarian on one hand, and a search engine providing automated access to a library catalog on another. The motivation was to study the reasons for the high rate of failure of on-line searches prevalent at that time, and translate this knowledge into a creating a more effective information retrieval (IR) system. The study revealed high similarity between the two types of interactions, in the types of queries as well as the way users posed their queries. The added fact that users did not use the advanced search engine features or actively reformulated their query indicated that users expected search engines to be effective surrogates of humans. Analysis of the types of errors users made while interacting with the system revealed that *semantic* errors contributed to more than 50% of failures. The rest were due to *lexical* or *syntactic* errors. With users unable or reluctant to convey their real information needs to the search engine, the lack of interaction that created trust and understanding, something the librarian provided, stood out. To this end, the study suggested measures like providing context, highlighting different aspects of the topic, development of a set of disambiguating questions, designing interfaces that stimulate browsing and conveying information about the corpus to the user.

Belkin et al. (1982a,b) proposed the Anomalous States of Knowledge (ASK) model, which emerged from the belief that the best-match principle followed in retrieval was fundamentally flawed. The best-match principle assumes that users can precisely specify their information needs using a set of terms – the same information need that they have a gap in knowledge about. In addition to this, there is an assumption of a functional equivalence between documents and information needs. While the best-match principle has been reasonably successful, it is prone to failure. The ASK model argues that the IR process is iterative and interactive, with channels of communication between the user and system to not only collaborate in the process but also allow for changes in the user's perception of the original problem. The model also contends that users can have different types of ASK, and the retrieval strategy and model need to adapt to those requirements. Casting IR as an information-seeking behavior, Belkin (1993) proposed that the central process was the users interaction with text and the central component of the system was the user. This view was shaped by the belief that IR is not a one-shot process but a series of interactions between the user and system to satisfy an information need that can change during the course of interaction.

Our development and analysis of interaction techniques is guided by the observations and suggestions from the studies and theories discussed above.

2.3 The utility of interaction

While much has been said about the need for user interaction, the question of whether it helps in practice is important. We are encouraged by ample evidence from past work on the utility of user interaction in improving search effectiveness. The most common form of interaction is document-level and term-level relevance feedback. This involves presenting users with a list of documents or selected terms from an initial search, and requesting them to mark those they believe are relevant to their information need. Using this input from the user, the initial query is modified to reflect the user's choices (Rocchio, 1966; Salton and Buckley, 1997). This procedure can also be done in an on-line fashion (Aalbersberg, 1992), i.e. the query is modified each time a selection is made. Starting with this simple model there have been multiple extensions and novel schemes that have been explored, and shown to be effective. Table 2.1 provides an overview of various explorations of interactive techniques in the past. More so, it has also been demonstrated by Koenemann and Belkin (1996) and Allan et al. (2001) that users prefer an interactive system to no interaction, and appreciate the opportunity to have some control over the search process. White and Ruthven (2006) provide empirical validation for this fact.

2.4 Predicting the utility of user interaction

Predicting when a query could potentially fail is a tough problem, and the focus of many efforts. Cronen-Townsend et al. (2002) used the relative entropy between the query language model and the collection language model to predict query difficulty. Their prediction experiments showed a positive correlation with the MAP of queries. He and Ounis (2004) extended this work to include a number of features in addition to relative entropy to predict query performance. These features include various metrics using inverse document frequencies (IDF) of the query terms, number of documents containing the query terms and the clarity score. Their investigations showed that different features were useful depending on the query length. Carmel et al. (2006) tried to understand the relationship between queries, the relevant documents and the collection in an effort to predict query difficulty. They concluded that the larger the (statistical) difference between the combination of the query and relevant documents from the corpus, the better the query's possibility of success. More recent work by Vinay et al. (2006) and Zhou and Croft (2006) has focused on measuring some or all of measures such as stability of the returned ranked list to query and document perturbation, clustering tendency in the ranked list, and local intrinsic dimensionality. In the context of this thesis, our goal is a bit different. We will develop features to determine if user interaction has the potential to improve performance on a query (Chapter 7), irrespective of its quality.

Reference	Summary
(Anick and Tipirneni, 1999)	Iterative query refinement though terminological feedback. Lexical compounds in result lists were shown to be a source of domain concepts that users can view and choose from easily.
(Cutting et al., 1992)	Users could browse by scattering documents into clusters or groups then gather a subset of these groups and re-scatter them to form new groups. Clusters were created based on content similarity between documents.
(White et al., 2005) (Tombros et al., 2003b)	More effective search was facilitated by displaying top-ranked sentences from returned documents in comparison to displaying lists of entire documents.
(Harman, 1988)	Selectively adding terms from an automatically created list of terms was shown to improve performance by 160%. Reported that background processes were very expensive.
(Hearst and Karadi, 1997)	Integrated search and browsing of very large category hierarchies with their associated text collections. Simultaneously displayed the representations of the categories and the retrieved documents. Displayed multiple selected categories simultaneously along with their hierarchical context. Developed a three-dimensional graphical workspace.
(Chen et al., 1998)	Extension of Scatter/Gather to simultaneous browsing and retrieval of web images and associated text
(Belkin et al., 1997)	Compared the performance and usability of a system offering positive relevance feedback with one offering positive and negative relevance feedback. Results were inconclusive about utility of negative feedback, but showed that both feedback systems were better than an automatic one.
(Leuski and Allan, 2000) (Leouski and Allan, 1998)	Described a user interface that integrated the ranked list with a clustering visualization. The visualization presented documents as spheres floating in space and positioned them in proportion to their inter-document similarity. User studies showed that users were more successful with the visualization than they were using a ranked list.

Table 2.1. Overview of some past work showing improvements in performance through user interaction utilizing various types of interfaces

2.5 Interaction strategies

Given the limitations of current automatic techniques, and the acknowledgment that IR is a complex process requiring a dialog between the user and the system, there is need for an intermediary (Borgman et al., 1988; Henninger and Belkin, 1996). This intermediary should help users identify the information they need, acquire the information, and formulate and direct search strategies. Abstracting away the actual design of the interface, there is still a need (Henninger and Belkin, 1996) to determine not only what information is required from the user but also the mode of interaction to get that information (White and Ruthven, 2006). Users prefer systems that gather information unobtrusively and wish to avoid actions like checking boxes as it distracts them from their goal of searching. Users also prefer to retain control over query creation, like having terms suggested to them, and appreciate the ability to verify the correctness of decisions with respect to retrieval strategies. In other words, users want to retain control over the strategic aspects of the IR process that directly influence the quality of results.

A majority of previous investigations into user interaction involved showing users textual information as evidence for feedback. The textual content varies from terms for expansion (Anick and Tipirneni, 1999; Harman, 1988), sets of terms to represent clusters (Cutting et al., 1992), top-ranked sentences (Tombros et al., 2003a,b; White et al., 2005), passages, and entire documents (Aalbersberg, 1992). Unlike this past work, in the main interaction techniques of query reduction and query expansion we explore in Sections 4.1.1 and 4.1.2, we not only present users options to choose from, but also provide them relevant information to understand the search space each option will take them to.

Researchers also used graphical means of eliciting information from users. These efforts were guided by the belief that graphical displays (Hearst, 1995; Hearst and Karadi, 1997; Hendley et al., 1995; Kumar et al., 1997; Leouski and Allan, 1998;

Swan and Allan, 2000, 1998) and images (Chen et al., 1998) were easier than text for users to process and navigate, and provided a quick summary of large amounts of information. Some work we propose on using images to summarize clusters of documents in a ranked list (Chapter 11) is motivated by such work.

A study by Dumais et al. (2001) explored the utility of showing context in different ways to guide users in judging relevance. The conclusion that category interfaces were more useful than list interfaces will have important implications on the perceptual and cognitive characteristics of the interfaces we design.

Croft and Thompson (1987) developed the I^3R system with the goal of providing a user support in the major phases of information retrieval namely query formulation and refinement, search and evaluation. The I^3R system was designed to act as an intelligent intermediary that guided the user through the search process and obtained feedback. Of particular interest to us is the availability of two retrieval strategies – a term-based probabilistic one and a cluster-based one. When the former failed, the latter was invoked. This is similar to our goal of invoking the right interaction strategy based on the query, except that we plan to automate the decision process and not expect the user to indicate that they were dissatisfied with a particular interactive session. The I^3R system also had a facility for user feedback that involved the user going through the final results and indicating which documents were relevant. We will not only automatically formulate search (interaction) strategies, but also provide the user succinct information about the type of information their query and its variants will retrieve. This is done during the query formulation stage itself.

The MERIT system developed by Belkin et al. (1995) views information retrieval as a process of information seeking, and that information seeking can be modeled as a dialog. These dialogs were characterized as a limited set of information seeking strategies. Belkin et al. (1995) proposed developing specific strategies i.e. dialog structures or scripts associated with each information seeking strategy to make the entire information retrieval process more efficient. However, their work falls short of developing techniques to automatically select scripts in response to a query, though it could be argued that the sort of information retrieval tasks they worked with were unsuited for such treatment. We too are interested in making user interaction more effective, and to that end are interested in developing strategies to select appropriate interaction techniques in response to an ad-hoc query.

2.6 Eliciting Information

Spink et al. (1996) conducted a study of elicitations, both by the user and intermediary, during human-mediated IR. The elicitations were mainly used to request information on search terms and strategies, selection of information sources, system output and relevance of retrieved materials, and users' background knowledge and expertise. To start with, it was observed that 64% of the elicitations were by the intermediary. The elicitations made by the intermediary occurred all through the search process, right from query formulation to result judgment. This was in contrast to elicitations from the user which were more during the online stage of the search interaction. Users became more active once an initial set of documents was retrieved. Their elicitations at this stage had to do with *cleaning-up* the results. Search intermediaries too were active more during the online interaction than at the beginning. This was because they had to create a good model of the user's information need using a set of terms.

The elicitation of information from the user can be done either through explicit or implicit means. For example, Kumaran and Allan (2006a,b) provided users a set of options with check boxes in their user interface. Users needed to explicitly select options to convey their response to the system. In a departure from the usual process of eliciting positive judgments from users Cool et al. (1996) asked users to indicate terms they thought were *not* useful in the context of the query. Kraft et al. (2006) had a different take on user interaction. This allowed the user to enter long queries like entire passages of text. This was a form of implicit feedback as the user automatically provided context along with the information need. Kelly et al. (2005) performed similar studies that showed that an interface that encouraged users to enter more terms describing their information need could gather better terms for use in a query. This was in comparison to automatic methods of query expansion. Other approaches to unobtrusively gather information from the user include tracking the time users spent viewing pieces of evidence like documents (Kelly and Belkin, 2004; Kelly and Teevan, 2003) and top-ranking sentences (White et al., 2005). The hypothesis behind this form of implicit feedback was that users will spend more time reading through relevant material than non-relevant material.

CHAPTER 3

DATA SETS AND EVALUATION

Results from studies by Borlund (2000) and Borlund and Ingwersen (1997) show that the practice of using 'simulated work task situations' to evaluate interactive IR is as effective as requiring users to use their own information needs. The 'simulated work task situation' involves the creation of an effective information-seeking scenario simulating a real information need. A description of the context/scenario of a given situation is used as a starting point for the participant's information need. The goal is to create a realistic setting where the participants understand the given description and develop individual and subjective information need interpretations. Participants are expected to then use the experimental system to assess relevance of the obtained results in relation to their perceptions of the information need and the underlying simulated work task situation. The user studies we will conduct using TREC[1] data sets are based on simulated work task situations.

3.1 Data Sets

As our data sets we used the TREC Robust 2004, Robust 2005 (Voorhees, 2006), TREC 5 ad-hoc (Voorhees and Harman, 1996) and HARD 2003 (Allan, 2003) document collections. The 2004 Robust collection contains around half a million documents from the Financial Times, the Federal Register, the LA Times, and FBIS. The Robust 2005 collection is the one-million document AQUAINT collection. The

[1] http://trec.nist.gov/

choice of Robust tracks was motivated by the fact that the associated queries were known to be difficult, and conventional IR techniques were known to fail for a number of them. The TREC 5 ad-hoc collection consists of TREC disks 1 and 2, and presented a standard ad-hoc retrieval setting. The HARD 2003 collection, a subset of the AQUAINT corpus and US government corpus containing 372,219 documents in all, was also selected since it was created for a track with focus on user interaction. The fifty queries in the Robust 05 data set overlap with those in the Robust 04 data set we used for training. However, since the collections are different, we limit the risk of over-fitting. The HARD data set uses the same collection as the Robust 04 data set, but has a different set of fifty queries. Finally, the TREC 5 data set shares neither the queries nor the collection with the Robust 04 data set. We believe that this choice of test data sets will provide a comprehensive validation of our techniques. 249 queries from the TREC Robust 2004 track were used to study the impact of the various techniques presented in this paper, and to learn parameters used for thresholding. The remaining 150 queries, 50 each from the three remaining tracks, were used to test the generality of our techniques.

All collections were stemmed using the Krovetz stemmer (Krovetz, 1993) provided as part of the Indri search engine (Section 3.3). We used a manually-created stop list of twenty terms (*a, an, and, are, at, as, be, for, in, is, it, of, on, or, that, the, to, was, with* and *what*).

3.2 Evaluation

Al-Maskari et al. (2007) have shown that measures based on cumulative gain (Järvelin and Kekäläinen, 2002) and precision correlate well with users' satisfaction of the results. Most research in Interactive IR uses mean average precision (MAP), precision at 5 documents (p@5), and precision at 10 documents (p@10) as the criterion to measure effectiveness. MAP is the most widely used measure in IR. While precision is the fraction of the retrieved documents that are relevant, average precision (AP) is a single value obtained by averaging the precision values at each new relevant document observed. MAP is the arithmetic mean of the average precisions of a set of queries. Similarly, GMAP (Robertson, 2006) is the geometric mean of the average precisions of a set of queries. The GMAP measure is more indicative of performance across an entire set of queries. MAP can be skewed by the presence of a few well-performing queries, and hence is not as good a measure as GMAP from the perspective of measure comprehensive performance. p@5 is the fraction of relevant documents in the top five retrieved documents while p@10 is the fraction in the top ten.

The measures we have discussed until now, with the exception of MAP, are oblivious to the *position* of the relevant documents in the ranked list. For example, a retrieval system can attain a p@5 value of 0.2 by retrieving a single relevant document anywhere in the top five in a ranked list. However, ranking the relevant document first is better than ranking it fifth. A measure that can also reflect the positional distribution of relevant documents is normalized discounted cumulative gain (NDCG).

NDCG was initially proposed by Järvelin and Kekäläinen (2002). The version we used is a slight variation described by Vassilvitskii and Brill (2006).

$$\text{NDCG} = N \sum_i \frac{2^{r(i)} - 1}{\log(1+i)}$$

where the summation is over all the documents retrieved for a query from a collection, $r(i)$ is the relevance of document in rank position i and $N$is a normalizing constant. The documents in the data sets we report results for have binary relevance judgments, i.e. $r(i) \in \{0, 1\}$. The log term in the denominator serves as a discounting factor that increases with the position of the result. Thus systems that return relevant documents higher up the ranked list have higher NDCG scores. In this thesis we report NDCG@15 scores, i.e. NDCG calculated based on the top fifteen documents returned by the retrieval system in response to a query.

However MAP, GMAP, p@5, p@10, and NDCG@15 do not provide a good measure of the amount of *effort* the user puts in during the interaction process. Kaki (2005) compared search effectiveness between using a query-based Internet search, a directory-based search and a phrase-based query reformulation assisted search. While the primary measure for comparison was *relevance rating*, a measure of the quality of the documents the subjects perused during search, *cognitive load* was also measured to determine the demands made on the user while interacting with the search mechanism. Cognitive load was measured by creating a secondary task called a digit-monitoring task. The task involved listening to a stream of digits and responding when a digit was repeated. By measuring the reaction time and miss rate in the secondary task, the amount of effort a subject employed was inferred based in the hypothesis that effort applied to the primary task was inversely proportional to the performance on the secondary task. Bruza et al. (1998) too used a secondary digit-monitoring task to determine the cognitive load imposed on users by query refinement mechanisms.

Further, Kaki (2004) proposed three *proportional* measures - *search speed*, *qualified search speed* and *immediate search accuracy*. Search speed is the number of answers found per minute of search time. Qualified search speed, while also measured in terms of answers per minute, tries to address the shortcomings of search speed by calculating the search speed on a per-relevance category basis. Thus, while search speed takes into account only the number of answers, qualified search speed takes into account the quality of the answers too. Thus we have

$$\text{search speed} = \frac{\text{answers found}}{\text{minutes searched}}$$

and

$$\text{qualified search speed}_{RCi} = \frac{\text{answers found}_{RCi}}{\text{minutes searched}}$$

where RCi stands for relevance category i. The typical categories are *relevant* and *non-relevant*.

Immediate accuracy is used to capture the success of typical web search behavior. In the web environment the typical goal for a search is to find just a few *good enough* answers to a question. Immediate accuracy is expressed as a success rate. The success rate is the proportion of cases where at least one relevant result is found by the n^{th} result inspected by the user. The selections for each task and user are gone through in the order they were made, and the frequency of first relevant result finding is calculated for each selection (first, second and so on). This sum is divided by the total number of observations (number of participants × number of tasks) to get the percentage of first relevant result found per selection. Formally,

$$\text{immediate accuracy}_n = \frac{\text{number of first relevant results}_n}{\text{total number of observations}}$$

Time as a measure of search speed and effectiveness has also been used in a number of studies (Dumais et al., 2001).

In this thesis, we evaluate the interactive techniques we explore by reporting MAP, GMAP, p@5, p@10, NDCG@15 and the time taken by users to complete search tasks when appropriate.

3.3 Search Engine

As our search engine we used version 2.6 of the Indri search engine (Strohman et al., 2005), developed as part of the Lemur[2] project. Indri's retrieval model (Metzler and Croft, 2004) combines the language modeling (Song and Croft, 1999) and inference network (Turtle and Croft, 1991) approaches to information retrieval. While the inference network-based retrieval framework of Indri permits the use of structured queries, the use of language modeling techniques provides better estimates of probabilities for query evaluation.

[2] http://www.lemurproject.org

CHAPTER 4

LONG QUERIES

In this chapter we introduce long queries, and the performance that can be achieved by using them in their original as well as modified forms. The modification can involve (a) dropping terms from the original query, i.e. query reduction or (b) adding appropriate terms to the original query, i.e. query expansion. We show the tremendous potential for improved performance that can be achieved if the right modifications are identified.

4.1 Long Queries

The Y!Q beta[1] search engine allows users to select large portions of text from documents and issue them as queries. The search engine is designed to encourage users to submit long queries such as this example from the web site *"I need to know the gas mileage for my Audi A8 2004 model"*. The motivation for encouraging this type of querying is that longer queries provide more information in the form of context (Kraft et al., 2006), and this additional information could be leveraged to provide a better search experience. However, handling such long queries is a challenge. The use of all the terms from the user's input can rapidly narrow down the set of matching documents, especially if a boolean retrieval model is adopted. While one would expect the underlying retrieval model to appropriately assign weights to different terms in the query and return only relevant content, it is widely acknowledged that models

[1] `http://yq.search.yahoo.com/`

fail due to a variety of reasons (Harman and Buckley, 2004b), and are not suited to tackle every possible query.

The queries used in the TREC ad-hoc tracks consist of title, description and narrative sections, of progressively increasing length. The title, of length ranging from a single term to four terms is considered a concise query, while the description is considered a longer version of the title expressing the same information need. Most research on the TREC ad-hoc retrieval track reports results using only the title portion as the query, and a combination of the title and description as a separate query. Most reported results show that the latter is more effective than the former, though in the case of some hard collections the opposite is true. However, as we shall show later, there is tremendous scope for improvement. Formulating a shorter query from the description can lead to significant improvements in performance. We now introduce the concepts of *query reduction* and *query expansion* as interpreted in this thesis.

4.1.1 Query Reduction

Consider the following query:

> *Define Argentine and British international relations.*

When this query was issued to a search engine, the average precision (AP) of the results was 0.424. When we selected subsets of terms (*sub-queries*) from the query, and ran them as distinct queries, the performance was as shown in Table 4.1. It can be observed that there are seven different ways of re-writing the original query to attain better performance. The best query *britain argentina*, also among the shortest, did not have a natural-language flavor to it, but had an effectiveness almost 50% more than the original query. This immense potential for improvement motivates further exploration of query reduction.

Query	AP
....	....
international relate	0.000
define international relate	0.000
....	....
define argentina	0.123
international relate argentina	0.130
define relate argentina	0.141
relate argentina	0.173
define britain international relate argentina	0.424
define britain international argentina	0.469
britain international relate argentina	0.490
define britain relate argentina	0.494
britain international argentina	0.528
define britain argentina	0.546
britain relate argentina	0.563
britain argentina	0.626

Table 4.1. The results of using all possible subsets (excluding singletons) of the original query as queries, in increasing order of MAP. The query terms were stemmed and stopped.

4.1.2 Query Expansion

Consider the following short (title) query for TREC Topic 370:

food drug law

When we expand this query with twenty-five terms obtained through pseudo-relevance feedback (PRF) (Lavrenko and Croft, 2001), we obtain an AP of 0.145 compared to an AP of 0.110 when the original query alone was used[2]. PRF involves building a language model of the vocabulary that is likely to occur in relevant documents. The initial query is used to rank documents and the top several documents are assumed to be relevant. The vocabulary of those documents is analyzed to calculate a probability distribution of words that are related to the query. The resulting prob-

[2] Expanding this short query makes it a long query.

Query	AP
....	....
product section act information under	0.038
product section act information under administrate	0.046
....	....
regulate product section food fda	0.395
regulate section food fda	0.399
regulate information under food fda	0.407
regulate product information under food fda	0.407
regulate product information food fda	0.415
regulate product under food fda	0.415
regulate product food fda	0.416
regulate information food fda	0.418
regulate under food fda	0.423
regulate food fda	0.430

Table 4.2. The results of using all possible subsets of the expansion terms suggested by pseudo-relevance feedback. The results are sorted in ascending order by AP. The original query *food drug law* was included with each expansion subset. The query terms were stemmed and stopped.

ability distribution is used as an additional component of the query with expanded vocabulary. PRF consistently improves retrieval performance over simpler language modeling approaches, and meets or beats other techniques based on automatic query expansion. However the choice of expansion terms selected by PRF is not always optimal. Frequently, PRF includes terms in the query that serve to only reduce its effectiveness. For example, if instead of just using all twenty-five terms for our example query, we considered all subsets and ran them as distinct queries, we observed that a large fraction of them performed much better than simple PRF. In Table 4.2 we can see that certain *expansion subsets* can lead to a 300% improvement in performance for this query. This motivates an exploration of techniques to automatically identify such expansion subsets.

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.472	0.397	0.379	0.240
PRF (Best)	*0.514*	*0.442*	*0.423*	*0.288*
Query Reduction				
Upper Bound (UB)	**0.799**	**0.671**	**0.626**	**0.366**

Table 4.3. The upper bound performance that can be achieved by identifying the best sub-query for 249 TREC 2004 Robust track *description* queries. Italicized values indicate that the scores are significantly better than the baseline, while those in bold are significantly better than PRF. Statistical significance was measured using a paired t-test, with α set to 0.05

4.2 Upper Bound Experiments

Tables 4.3 and 4.4 shows the best performance that can be achieved through query reduction for long queries and query expansion for short queries respectively. All results are reported for queries from the TREC Robust 2004 track. We treated the description portion of TREC queries as long queries for our experiments, and the titles as short queries. Baseline refers to a query-likelihood (QL) run using the Indri search engine, while PRF refers to automatic query expansion using pseudo-relevance feedback[3]. "Upper Bound" refers to the situation when the best sub-query and best expansion subset was used for query reduction and expansion respectively. In other words, if we had access to an oracle that always provided us the best sub-query and best expansion set for a query, we can obtain the indicated upper bound on performance.

4.3 Summary

The P@5, P@10, NDCG@15, and MAP values in Tables 4.3 and 4.4 for query reduction and expansion show the tremendous scope for improvement in the case of

[3] All performance metrics reported for PRF are upper bounds. We performed comprehensive parameter sweeps for each collection reported in this paper to determine the best parameter settings. The performance in practice will be lower as sub-optimal parameters will be learned for each collection by training on other collections.

System	P@5	P@10	NDCG@15	MAP
Baseline (PRF)	0.462	0.420	0.433	0.284
Query Expansion				
Upper Bound (UB)	**0.696**	**0.600**	**0.570**	**0.352**

Table 4.4. The upper bound performance that can be achieved by identifying the best expansion subset for 249 TREC 2004 Robust track *title* queries. Values in bold are significantly better than PRF, which was performed considering the top 25 documents and expanding by 20 terms. Statistical significance was measured using a paired t-test, with α set to 0.05

information retrieval using long queries. The upper bounds can be viewed as targets for any techniques we develop for query reduction and expansion. The degree of improvement progressively increases for the MAP, NDCG@15, P@10 and P@5 measures, indicating that the techniques of query reduction and expansion are precision-enhancing. The potential 70% and 50% improvements in P@5 through reduction and expansion are unlike those that can be realized through contemporary automatic techniques, and require the exploration of new automatic and interactive techniques which are the focus of this thesis. In the next chapter we will describe techniques to automatically reformulate long queries targeting the upper bounds we have identified so far.

CHAPTER 5
AUTOMATIC TECHNIQUES

In the previous chapter we showed that significant improvements in performance can be obtained if we can identify the best sub-queries and expansion subsets. We can also view this as the problem of identifying the key terms or concepts in long queries. In this chapter we will describe the techniques we investigated to do so automatically. After identifying the best sub-query and expansion subset for the set of 249 training queries, we compared the terms retained in them with those in the original query. We made the following observations that informed techniques to identify good sub-queries and expansion subsets.

1. Terms in the original query that a human would consider vital to convey the type of information desired were missing from the best sub-queries. For example, the best sub-query for the example in Section 4.1.1 was *britain argentina*, omitting any reference to international relations. This also reveals a mismatch between the user's query and the way terms occurred in the corpus, and suggests that an approximate query could at times be a better starting point for search.

2. The sub-query would often contain *only* terms that a human would consider vital to the query while the original query would also (naturally) contain them, albeit weighted lower with respect to other terms. This is a common problem (Harman and Buckley, 2004b), and the focus of efforts to isolate the key *concept* terms in queries (Allan et al., 1996; Buckley et al., 2000).

3. Good sub-queries were missing many of the noise terms found in the original query. Ideally the retrieval model would weight them lower, but dropping them completely from the query appeared to be more effective.

4. Sub-queries a human would consider as an incomplete expression of information need sometimes performed better than the original query. Our example illustrates this point.

5. Smaller-sized expansion subsets led to higher gains in performance. A few key terms were enough to boost performance; more terms only reduced the quality of the query.

6. Terms that would ordinarily be regarded as stop words sometimes proved more useful than *content* words.

Given the above empirical observations, we now explore a variety of procedures to automatically adapt the system to the user's query. We expect that a good query would have the following properties.

A. *Minimal Cardinality*: Any set that contains more than the minimum number of terms to retrieve relevant documents could suffer from concept drift.

B. *Coherency*: The terms that constitute the sub-query should be coherent, i.e. they should buttress each other in representing the information need. If need be, terms that the user considered important but led to retrieval of non-relevant documents should be dropped.

5.1 Automatic Selection Techniques

Our goal is to identify a subset of the original query (for query reduction) or expanded set (for expansion). We hypothesize that such a subset needs to be statistically *cohesive*, i.e. all the terms support the retrieval of only relevant material, and focus the query on a (single) relevant portion of the search space. To find such a subset we scored the entire universe of subsets using a measure based on co-occurrence of terms constituting each subset, and selected the one with the best score. Some of the earliest work on the use of statistical co-occurrence of terms for improving search queries were by Doyle (1959, 1975) and Maron and Kuhns (1960). Stiles (1961) and Rijsbergen (1979) suggested the use of dependence trees, i.e. trees with terms as vertices and edges weighted by a measure of association between the terms, to capture the most significant dependencies between terms in a query. While Stiles (1961) explored the use of the χ^2 to measure association between query terms, Rijsbergen (1979) used the expected mutual information measure. In this thesis we use the mutual information measure (see below). Our choice was prompted by the large amount of evidence in past work of the utility of this measure in text processing and retrieval tasks. We acknowledge that other measures could potentially work as well as if not better than mutual information, but leave that exploration for future work.

5.1.1 Mutual Information

Let X and Y be two random variables, with joint distribution $P(x, y)$ and marginal distributions $P(x)$ and $P(y)$ respectively. The mutual information is then defined as:

$$\begin{aligned} I(X;Y) &= \sum_x \sum_y p(x,y) log \frac{p(x,y)}{p(x)p(y)} \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{5.1}$$

$H(X)$ and $H(Y)$ are marginal entropies, $H(Y|X)$ and $H(X|Y)$ are conditional entropies, and $H(X, Y)$ is the joint entropy. Intuitively, mutual information measures the information about X that is shared by Y. If X and Y are independent, then X contains no information about Y and vice versa and hence their mutual information is zero. Mutual Information is attractive because it is not only easy to compute, but also takes into consideration corpus statistics and semantics. We calculated the mutual information between two terms using Equation 5.2, described in Church and Hanks (1990).

$$I(x,y) = log\frac{\frac{n(x,y)}{N}}{\frac{n(x)}{N}\frac{n(y)}{N}} \tag{5.2}$$

where $n(x,y)$ is the number of times terms x and y occurred within a term window of 100 terms across the corpus, while $n(x)$ and $n(y)$ are the frequencies of x and y in the collection of size N terms.

To tackle the situation where we have an arbitrary number of variables (terms) we extend the two-variable case to the multivariate case. The extension, called multivariate mutual information (MVMI) can be generalized from Equation 5.1 to:

$$I(X_1;X_2;X_3;...;X_N) = \sum_{i=1}^{N}(-1)^{i-1}\sum_{X\subset(X_1,X_2,X_3,...,X_N),|X|=k} H(X) \tag{5.3}$$

The calculation of multivariate information using Equation 5.3 was very cumbersome, and we instead worked with the approximation of Kern et al. (2003) given below.

$$I(X_1;X_2;X_3;...;X_N) = \sum_{i,j=\{1,2,3,...,N;i\neq j\}} I(X_i;X_j) \tag{5.4}$$

For the case involving multiple terms, we calculated MVMI as the sum of the pair-wise mutual information for all terms in the candidate sub-query. This can be also viewed as the creation of a completely connected graph $G = (V,E)$, where the vertices V are the terms and the edges E are weighted using the mutual information between the vertices they connect.

To select a score representative of the quality of a sub-query or expansion subset we considered several options including the sum, average, median and minimum of the edge weights. We performed experiments on a set of candidate queries to determine how well each of these measures tracked AP, and found that the average worked best. We refer to the selection procedure using the average score as *Average*.

5.1.2 Maximum Spanning Tree

It is well-known that an average is easily skewed by outliers. In other words, the existence of one or more terms that have low mutual information with every other term could potentially distort results. This problem could be further compounded by the fact that mutual information measured using Equation 5.2 could have a negative value. Previous work by Rijsbergen (1979) showed that the Maximum Spanning Tree (MaxST) over the fully connected graph G captured the most significant of the dependencies between query terms. Based on this result, we used the weight of the MaxST as a measure of the candidate query's quality. We used Kruskal's minimum spanning tree algorithm (Cormen et al., 2001) after negating the edge weights to obtain a MaxST. We refer to the selection procedure using the weight of the maximum spanning tree as *MaxST*.

5.1.3 Named Entities

Named entities (names of persons, places, organizations, dates, etc.) are known to play an important anchor role in many information retrieval applications. In our example for query reduction, sub-queries without *Britain* or *Argentina* will not be effective even though the mutual information score of the other two terms *international* and *relations* might indicate otherwise. We experimented with another version of sub-query selection that considered only sub-queries that retained at least one of the named entities from the original query. We refer to the variants for query reduction that retained named entities as *NE_Average* and *NE_MaxST*. For query expansion, we did not pursue expansion by only named entities.

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.472	0.397	0.379	0.240
Upper Bound (UB)	**0.799**	**0.671**	**0.626**	**0.366**
Average	0.304	0.268	0.261	0.172
MaxST	0.300	0.264	0.259	0.171
NE_Average	0.303	0.264	0.261	0.170
NE_MaxST	0.315	0.274	0.273	0.180

Table 5.1. Performance when the highest ranked sub-query selected by various techniques was used. Results are based on 249 description queries from TREC Robust 2004. Values in bold are significantly better than the baseline. Statistical significance was measured using a paired t-test, with α set to 0.05

5.1.4 Experimental Setup

We used version 2.6 of the Indri search engine. The pseudo-relevance feedback technique we used was based on relevance models (Lavrenko and Croft, 2001). More details about the experimental setup are available in Section 3.1.

To extract named entities from the queries, we used BBN Identifinder (Bikel et al., 1999). The named entities identified were of type *Person*, *Location*, *Organization*, *Date*, and *Time*.

5.1.5 Results

To evaluate the automatic sub-query or expansion subset selection procedures developed we performed retrieval experiments using the queries selected by them. For query reduction (Table 5.1), the results of automatic selection were worse than even the baseline, and there was no significant difference between using any of the different sub-query selection procedures with the exception of NE_MaxST.

Similar results were observed for query expansion (Table 5.2). While there was an improvement in the P@5 values, the MAP and NDCG@15 measures dropped.

Clearly, the automatic techniques based on the features and techniques we selected were not successful in identifying the best sub-queries and expansion subsets that would lead to the performance presented in Tables 4.3 and 4.4.

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.462	0.420	0.433	0.284
Upper Bound (UB)	**0.696**	**0.600**	**0.570**	**0.352**
Average	0.481	0.423	0.396	0.259
MaxST	0.479	0.426	0.401	0.263

Table 5.2. Performance when the highest ranked expansion subset selected by various techniques was used. Results are based on 249 title queries from TREC Robust 2004. Values in bold are significantly better than the baseline. Statistical significance was measured using a paired t-test, with α set to 0.05

5.2 The Utility of Ranking Sub-queries and Expansion Subsets

The limited utility of the automatic techniques could be attributed to the fact that we were working with the assumption that a procedure designed to favor term co-occurrence could be used to model a user's information need. To see if there was any general utility in using the procedures to select sub-queries, we viewed the procedures not as a way to select the best sub-query or expansion subset but as a way to *rank* all of them. Once we ranked all the sub-queries and expansion subsets, we selected the best-performing one from the top ten ranked by each selection procedure. While the effectiveness in each case for query reduction as measured by MAP was not close to the best possible MAP, 0.366, they were all significantly better than the baseline of 0.240 (Table 5.3,"Best Performance"). Similarly, in the case of query expansion (Table 5.4,"Best Performance") we notice improvements ranging from 22% in P@5 to 4% in MAP. However, if we select randomly from the top ten sub-queries in the case of query reduction or expansion subsets in the case of query expansion, we notice that on average ("Average Performance" in Tables 5.3 and 5.4) the performance is significantly worse than "Best Performance". Further, if we always select the worst sub-query and expansion subset ("Worst Performance" in Tables 5.3 and 5.4), performance plummets in the case of query reduction, and reduces significantly in the case of query expansion.

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.472	0.397	0.379	0.240
Upper Bound (UB)	**0.799**	**0.671**	**0.626**	**0.366**
Best Performance				
AverageTop10	**0.625**	**0.515**	**0.490**	**0.294**
MaxSTTop10	**0.633**	**0.527**	**0.498**	**0.300**
NE_AverageTop10	**0.610**	**0.510**	**0.482**	**0.291**
NE_MaxSTTop10	**0.634**	**0.527**	**0.499**	**0.302**
Average Performance				
AverageTop10	0.268	0.244	0.259	0.161
MaxSTTop10	0.302	0.275	0.292	0.183
NE_AverageTop10	0.281	0.255	0.270	0.169
NE_MaxSTTop10	0.319	0.289	0.309	0.194
Worst Performance				
AverageTop10	0.040	0.039	0.041	0.030
MaxSTTop10	0.027	0.023	0.027	0.014
NE_AverageTop10	0.072	0.073	0.074	0.051
NE_MaxSTTop10	0.040	0.047	0.051	0.031

Table 5.3. "Best Performance" refers to the performance when the highest ranked sub-query selected by various techniques was used. "Average Performance" refers to the situation when sub-queries are selected at random, while "Worst Performance" is a lower bound where the worst sub-query is always selected. Statistical significance was measured using a paired t-test, with α set to 0.05. We can notice that all sub-query selection techniques deliver "Best Performance" results better than the baseline with respect to all measures

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.462	0.420	0.433	0.284
Upper Bound (UB)	**0.696**	**0.600**	**0.570**	**0.352**
Best Performance				
AverageTop10	**0.564**	**0.499**	**0.457**	**0.298**
MaxSTTop10	**0.554**	**0.493**	**0.454**	**0.296**
Average Performance				
AverageTop10	0.418	0.379	0.390	0.258
MaxSTTop10	0.426	0.387	0.398	0.263
Worst Performance				
AverageTop10	0.353	0.330	0.314	0.207
MaxSTTop10	0.345	0.315	0.306	0.195

Table 5.4. "Best Performance" refers to the performance when the highest ranked expansion subset selected by various techniques was used. "Average Performance" refers to the situation when expansion subsets are selected at random, while "Worst Performance" is a lower bound where the worst expansion subset is always selected. Values in bold are significantly better than the baseline. Statistical significance was measured using a paired t-test, with α set to 0.05

Thus, if we can identify the best option from the top ten ranked by the various techniques we have described in this chapter, then significant improvements in performance are possible. The significant drop in performance in the average and worst cases imply that identifying the best option is very difficult, and mistakes in doing so can be very expensive.

From the upper bound or "best performance" results it appears that the automatic techniques we have developed are successful in biasing the better query reformulations higher up the ranked list. To confirm this hypothesis, we compared the distribution of MAPs when sub-queries were randomly sampled from exponential number available, and randomly sampled from the top ten ranked by the NE_MaxST technique.

Query Reduction: Figure 5.1 provides the distribution of MAPs of samples from the entire set of combinations of query terms and the set of top ten combinations obtained by the NE_MaxST ranking procedure for query reduction. The reason for selecting the NE_MaxST ranking procedure is provided in Section 6.1.1. A two-tailed

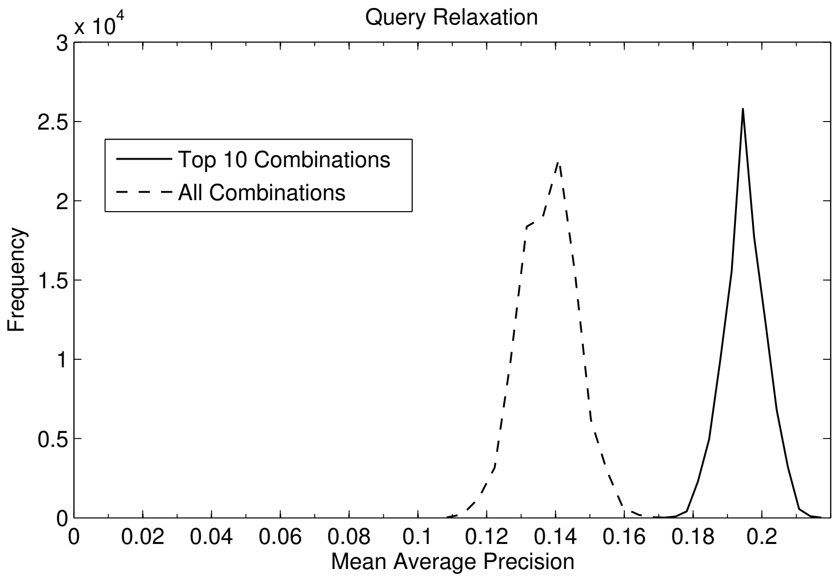


Figure 5.1. Comparison of the distributions of MAPs of all subsets of query terms and combinations selected by the NE_MaxST ranking procedure. The highest achievable MAP by always selecting the best from the top ten combinations is 0.302

paired t-test revealed that the distributions are significantly different ($\alpha = 0.05, p = 0$).

Greater improvements over the baseline at P5 and P10 were observed for query reduction (Table 5.3). While this is an indicator that the technique is precision-enhancing, it doesn't show whether these improvements were for already well-performing queries or otherwise. Figure 5.2 shows the distribution of the number of relevant documents in the top ten in the ranked list after simulated query reduction, i.e. when the best option from the top ten ranked by the NE_MaxST technique is selected. The distributions are for queries that had zero, one, and two relevant documents in the top ten documents retrieved by the baseline system. We can observe that greater numbers of relevant documents are present in the top ten documents in the ranked list for the queries that exhibited poor baseline performance.

We also investigated whether the success of query reduction had a correlation with the number of relevant documents for each query. The goal was to check if

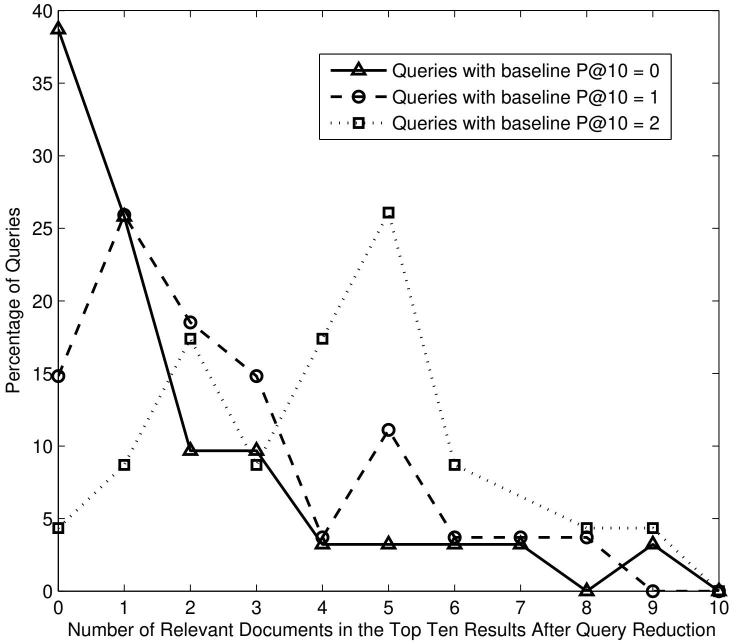


Figure 5.2. Distribution of number of relevant documents in the top ten in the ranked list after query reduction. The queries chosen were those with baseline P10 = 0, 1, and 2.

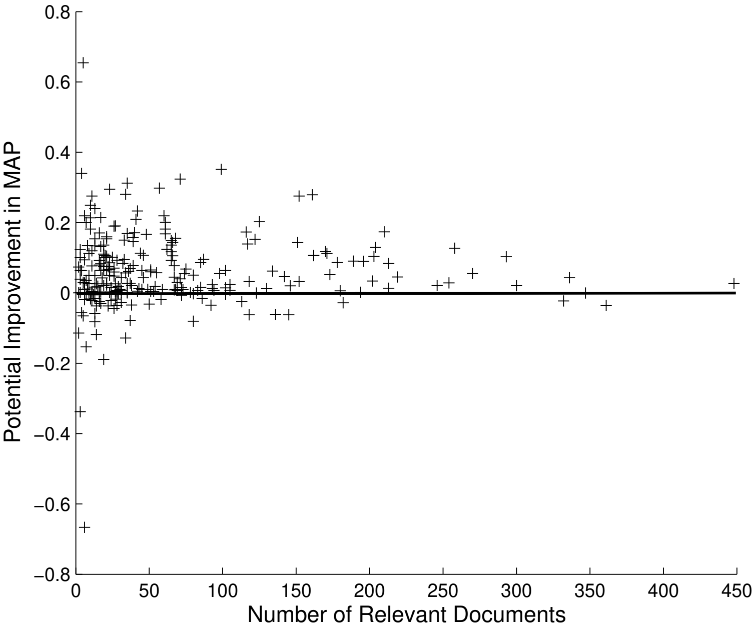


Figure 5.3. Scatter plot of number of relevant documents versus potential improvement in MAP over the baseline due to query reduction. Potential MAP due to query reduction refers to the case when the best sub-query from the top ten ranked by the NE_MaxST technique was selected for each query. Each query in our training set of 249 Robust 2004 queries had 66 relevant documents on average.

there was a positive correlation between success of query reduction and number of relevant documents. A positive correlation would indicate that queries with more relevant documents were better suited for query reduction. However, as Figure 5.3, a scatter plot of the potential MAP improvement due to query reduction for our training queries and the number of relevant documents for each query, shows there is hardly any correlation between the two. The correlation coefficient of 0.003 conveyed almost complete independence.

Query Expansion: Figure 5.4 illustrates the distribution of MAPs of samples from the entire set of combinations of expansion terms and the set of top ten combinations. A two-tailed paired t-test again shows that the distributions are significantly different ($\alpha = 0.05, p = 0$).

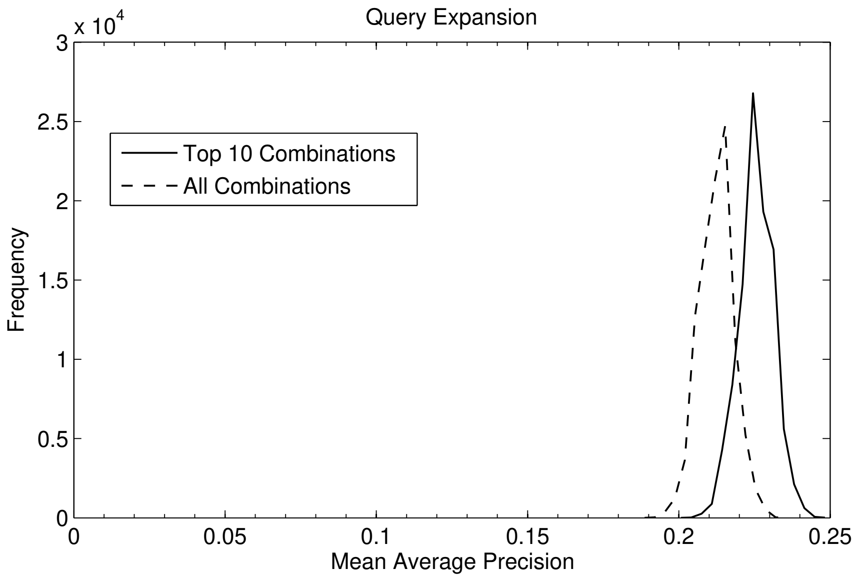


Figure 5.4. Comparison of the distributions of MAPs of all combinations of query expansion terms and combinations selected by the NE_MaxST ranking procedure. The highest achievable MAP by always selecting the best from the top ten combinations is 0.294

5.3 Summary

We have described techniques for ranking sub-queries and expansion subsets in such a way that better combinations are ranked higher. With these results are a background, the next step is to identify the best sub-query or expansion subset from a set of top-ranked combinations. Our approach is to involve the user in selecting the best from the top ten ranked by the automatic techniques described in this chapter. In the next chapter we will show how automatic techniques for ranking sub-queries and expansion subsets in concert with user interaction can be used to improve retrieval performance.

CHAPTER 6
FEASIBILITY STUDY

The final results we presented in the last chapter hinted at a potential for user interaction in the form of interactive query reduction (IQR) and interactive query expansion (IQE). We envisioned providing users with a list of the top ten options identified using a ranking procedure, and asking her to select the option she felt was most appropriate. We will refer to sub-queries and expansion subsets collectively as *options*. This additional round of human intervention could potentially compensate for the inability of the selection techniques described in Chapter 5 to select the best sub-query or expansion subset automatically. To this end we conducted a feasibility study to determine if users can select an option better than the baseline give a set of top ten options generated using the automatic techniques described in the previous chapter.

In the next section we will discuss issues related to the design of the user interface used for the feasibility study. In Section 6.2 we will present results of the feasibility study we conducted for IQR and IQE, and show that users can

- Use the supporting information provided in the interface to select a better reformulation of the original query.

- Use the same information to decide that none of the options presented were useful for some queries.

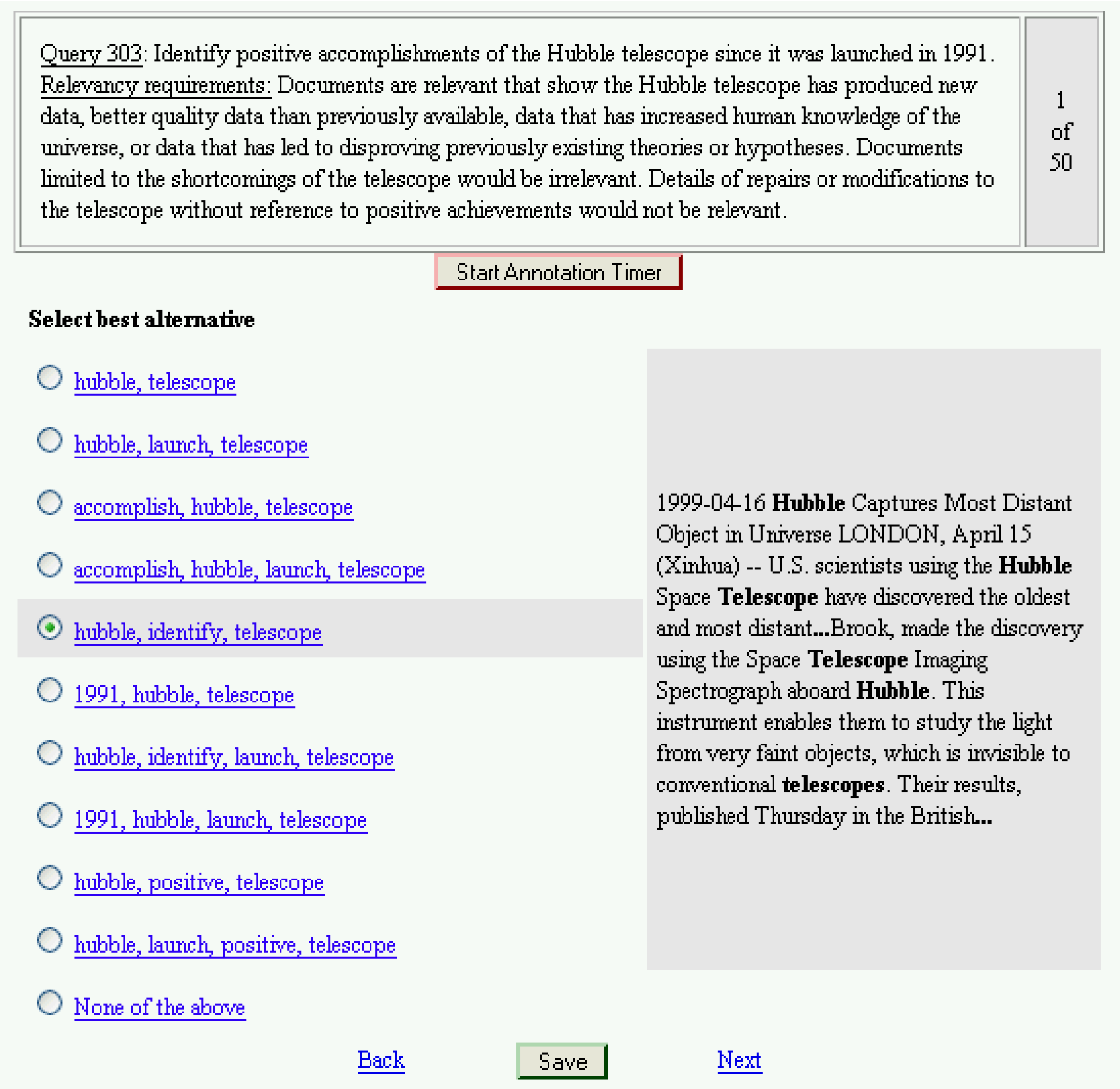
- Obtain significant improvements in performance through IQR and IQE.

In Section 6.3 we report the profiles of the feasibility study participants, and present the feedback on IQR and IQE obtained from them. We will present conclusions in Section 6.4, and discuss the limitations of the feasibility study we conducted in Section 6.5.

6.1 User Interface Design

The interface was designed keeping in mind the need for minimal interaction. Our goal was to show not only the ten options to users, but also provide some information on the result of selecting each of those options. Drawing from previous conclusions in the literature (Dumais et al., 2001; White et al., 2005) we decided to show users a snippet of text from the top-ranking document retrieved by each option in a tabbed interface. Figure 6.1 is a screen shot of the interface we provided to users to guide the system's adaptation to queries.

For IQR, we displayed the description (the *long* query) portion of each TREC query in the interface. The narrative was also included to help the user understand the information need. The title was kept hidden to avoid influencing the participant's choice of the best sub-query. For IQE, the roles of title and description were interchanged. A list of options was displayed along with links that could be clicked on to display a short section of text, or snippet, in a designated area. The intention was to provide an example of what would potentially be retrieved with a high rank if an option was selected. The user was expected to use this information to select the best option from the list. In situations where users observed multiple options retrieving the same snippet they were instructed to select the most general option. A facility to indicate that none of the options were good was also included.



Query 303: Identify positive accomplishments of the Hubble telescope since it was launched in 1991. Relevancy requirements: Documents are relevant that show the Hubble telescope has produced new data, better quality data than previously available, data that has increased human knowledge of the universe, or data that has led to disproving previously existing theories or hypotheses. Documents limited to the shortcomings of the telescope would be irrelevant. Details of repairs or modifications to the telescope without reference to positive achievements would not be relevant.

1 of 50

Select best alternative

- ○ hubble, telescope
- ○ hubble, launch, telescope
- ○ accomplish, hubble, telescope
- ○ accomplish, hubble, launch, telescope
- ● hubble, identify, telescope
- ○ 1991, hubble, telescope
- ○ hubble, identify, launch, telescope
- ○ 1991, hubble, launch, telescope
- ○ hubble, positive, telescope
- ○ hubble, launch, positive, telescope
- ○ None of the above

1999-04-16 **Hubble** Captures Most Distant Object in Universe LONDON, April 15 (Xinhua) -- U.S. scientists using the **Hubble** Space **Telescope** have discovered the oldest and most distant**...**Brook, made the discovery using the Space **Telescope** Imaging Spectrograph aboard **Hubble**. This instrument enables them to study the light from very faint objects, which is invisible to conventional **telescopes**. Their results, published Thursday in the British**...**

Figure 6.1. Screenshot of the annotation interface. This particular example is for query reduction.

	Percentage of candidates better than baseline
Average	28.5%
MaxST	35.5%
NE_Average	31.1%
NE_MaxST	36.6%

Table 6.1. Number of candidates from top 10 that exceeded the baseline

6.1.1 User interface content issues

The two key issues we faced while determining the content of the user interface were:

A. *Deciding which sub-query selection procedure to use to get the top 10 candidate sub-queries:* To determine this in the absence of any significant difference in performance due to the top-ranked candidate selected by each procedure, we looked at the number of candidates each procedure brought into the top 10 that were better than the baseline query, as measured by MAP. This was guided by the belief that the greater the number of better candidates in the top 10, the higher the probability that the user would select a better sub-query. Table 6.1 shows how each of the selection procedures compared. The NE_MaxST ranking procedure had the most number of better sub-queries in the top 10, and hence was chosen. For query expansion, we choose MaxST.

B. *Displaying context:* Simply displaying a list of 10 candidates without any supportive information would make the task of the user difficult. This was in contrast to query expansion techniques (Anick and Tipirneni, 1999) where displaying a list of terms sufficed as the task of the user was to disambiguate or expand a short query. An experiment was performed in which a single user worked with a set of 30 queries from Robust 2004, and an accompanying set of 10 candidate sub-queries each, twice - once with passages providing context and one with snippets providing context. The top-ranked passage was generated by modifying the candidate query into one that retrieved passages of fixed length instead of documents. Snippets, like those seen along

	MAP	GMAP
Snippet as Context	0.348	0.170
Passage as Context	0.296	0.151

Table 6.2. Results showing the MAP over 19 of 30 queries that the user provided selections for using each context type.

with links to top-ranked documents in the results from almost all popular search engines, were generated after a document-level query was used to query the collection. The order in which the two contexts were presented to the user was randomized to prevent the user from assuming a quality order. We see that presenting the snippet led to better MAP than presenting the passage (Table 6.2). The reason for this could be that the top-ranking passage we displayed was from a document ranked lower by the document-focused version of the query. Since measures like MAP and NDCG@15 are sensitive to the position of relevant documents, and the snippet was generated from the top-ranked document, we hypothesize that this led to the snippet being a better context to display.

6.2 Feasibility Study

We conducted an exploratory feasibility study with twelve participants that were a mix of volunteers and paid annotators. The participants were tasked with selecting the best option from a list of ten provided for query reduction and expansion. They were asked to base their decision on the snippet of text that corresponded to each option. To measure the time it took to complete the task for every query, we instructed the participants to start a timer after they read and understood the query, and just before they started inspecting the options. We used fifty queries from the Robust 2005 track (Voorhees, 2006) for this study. For query reduction, we used the description portion of each query, and for expansion the title. The baseline for query expansion was a PRF run with number of feedback documents and terms set to fifteen and twenty respectively.

6.2.1 Query Reduction

Table 6.3 shows that all participants were able to choose sub-queries that led to an improvement in performance over the baseline (description query). This improvement is not only in MAP but also in GMAP, indicating that user interaction helped improve a wider set of queries. Most notable were the improvements in P@5 and P@10. We believe that this was due to the fact that the information participants used for guidance was a snippet from the top-ranked document for each sub-query. Selecting an option implied that the participant automatically ensured a relevant document was retrieved in the first position of the ranked list. The interaction technique we developed was thus precision-enhancing. Another interesting result, from *# sub-queries selected* was that participants were able to decide in a large number of cases that re-writing was either not useful for a query, or that none of the options presented to them were better. Showing context appears to have helped. The average time taken by the participants to select an option was a minute and a half. It is important to note that the values of various measures reported in Table 6.3 are not comparable across users. This is because we have reported scores for each user based on the queries for which the user selected an option. From Table 6.3, we can see that users selected options for different numbers of queries.

6.2.2 Query Expansion

Table 6.4 summarizes the results of our study to evaluate the utility of IQE. Only 50% of participants achieved an improvement in either MAP or GMAP over the baseline expanded title query. Almost all of them however achieved improvements in P@5 and P@10, a trend noticed in the case of IQR too. This again attested to the fact that the interaction technique we utilized was precision-enhancing. Given that PRF already constitutes a very competitive baseline, and given the relatively little room for improvement (see *Upper Bound* in each case), we believe it is encouraging

	# Sub-queries selected	Selections above baseline	Avg. Time per Query (seconds)		MAP	GMAP	P@5	P@10
1	32	17	125	Baseline	0.176	0.108	0.463	0.422
				With Interaction	0.191	0.103	0.444	0.419
				Upper bound	0.239	0.150	0.538	0.519
2	35	20	107	Baseline	0.175	0.118	0.463	0.426
				With Interaction	0.196	0.124	0.469	0.431
				Upper bound	0.256	0.185	0.554	0.546
3	42	19	9	Baseline	0.175	0.107	0.443	0.417
				With Interaction	0.179	0.104	0.476	0.431
				Upper bound	0.251	0.173	0.529	0.517
4	28	19	91	Baseline	0.179	0.126	0.507	0.454
				With Interaction	0.205	0.143	0.586	0.525
				Upper bound	0.273	0.209	0.621	0.607
5	44	20	53	Baseline	0.173	0.105	0.445	0.418
				With Interaction	0.186	0.000	0.414	0.398
				Upper bound	0.234	0.127	0.491	0.491
6	34	17	28	Baseline	0.181	0.124	0.459	0.426
				With Interaction	0.228	0.155	0.535	0.541
				Upper bound	0.262	0.202	0.547	0.556
7	31	18	75	Baseline	0.185	0.123	0.471	0.439
				With Interaction	0.209	0.135	0.516	0.471
				Upper bound	0.268	0.198	0.587	0.574
8	34	20	92	Baseline	0.168	0.113	0.447	0.415
				With Interaction	0.206	0.143	0.512	0.485
				Upper bound	0.248	0.180	0.529	0.532
9	36	20	131	Baseline	0.191	0.134	0.478	0.450
				With Interaction	0.196	0.117	0.500	0.458
				Upper bound	0.278	0.206	0.600	0.594
10	37	15	22	Baseline	0.169	0.102	0.416	0.403
				With Interaction	0.211	0.128	0.470	0.459
				Upper bound	0.246	0.140	0.486	0.503
11	28	12	50	Baseline	0.168	0.113	0.464	0.439
				With Interaction	0.189	0.114	0.400	0.418
				Upper bound	0.216	0.148	0.514	0.511
12	29	12	75	Baseline	0.197	0.148	0.490	0.466
				With Interaction	0.235	0.162	0.531	0.541
				Upper bound	0.294	0.236	0.614	0.617

Table 6.3. IQR: All participants worked through a set of fifty queries. *# Sub-queries selected* refers to the number of queries for which the participant chose an option. *Selections above baseline* refers to the number of times the option selected by the user was better than the baseline query. All scores, including upper bounds, were calculated only considering the queries for which the participant selected a sub-query.

that participants registered improved performance over the baseline. We noticed from *# sub-queries selected* that participants decided in a larger number of cases that expansion was either not useful for a query, or that none of the options presented to them were better. The time taken by the participants to go through the options, read through the snippets and select an option varied from approximately half to one minute. This was much less that the times observed for IQR. The reason was that in the case of IQE a large number of options retrieved the same snippet resulting in the annotators having to read through less text before selecting an option. We will use this observation to reduce the number of options presented to users in Chapter 8.

6.3 User Profiles and Feedback

Our user pool consisted of twelve individuals, of which nine were male and three were female. The average age was 25.9 years, with eight of the users having Masters degrees and four of them with Bachelors. The number of queries that each of them issued to commercial search engines varied from ten to fifty each day.

Ten of the users considered the IQE task easier than the IQR task, and conveyed that a large number of IQE options had the same associated snippet, reducing the amount of text that had to be read. We will use this observation in Chapter 8 to reduce the number of options shown to users. All users were unanimous in their opinion that the interface was easy to use, but had mixed responses to the question "Would you like a commercial search engine to use such an interface?". Given below are the responses of some of the users who agreed to provide a reason.

1. *No. It is not very convenient for me to put a mouse upon the words and then see the whole snippets.*

2. *YES, as it helps to clarify the query. However the quality of the extracted passages should be improved*

	# Expansion sub-sets selected	Selections above baseline	Avg. Time per Query (seconds)		MAP	GMAP	P@5	P@10
1	30	15	57	Baseline	0.333	0.256	0.553	0.587
				With Interaction	0.335	0.249	0.587	0.597
				Upper bound	0.347	0.264	0.600	0.603
2	24	16	54	Baseline	0.367	0.312	0.675	0.683
				With Interaction	0.363	0.298	0.708	0.675
				Upper bound	0.371	0.306	0.700	0.679
3	13	8	67	Baseline	0.313	0.241	0.646	0.646
				With Interaction	0.311	0.244	0.677	0.646
				Upper bound	0.335	0.259	0.662	0.662
4	22	14	53	Baseline	0.348	0.293	0.645	0.677
				With Interaction	0.362	0.315	0.682	0.714
				Upper bound	0.351	0.299	0.636	0.682
5	32	20	57	Baseline	0.324	0.246	0.581	0.597
				With Interaction	0.313	0.221	0.594	0.609
				Upper bound	0.329	0.243	0.600	0.616
6	30	17	30	Baseline	0.314	0.246	0.560	0.573
				With Interaction	0.305	0.220	0.567	0.583
				Upper bound	0.322	0.241	0.593	0.597
7	22	11	29	Baseline	0.353	0.279	0.664	0.673
				With Interaction	0.356	0.277	0.682	0.695
				Upper bound	0.371	0.296	0.709	0.705
8	22	13	26	Baseline	0.357	0.267	0.673	0.705
				With Interaction	0.374	0.323	0.745	0.773
				Upper bound	0.379	0.291	0.709	0.745
9	16	8	28	Baseline	0.310	0.253	0.650	0.662
				With Interaction	0.314	0.244	0.737	0.694
				Upper bound	0.319	0.241	0.675	0.656
10	24	13	20	Baseline	0.383	0.329	0.717	0.729
				With Interaction	0.406	0.349	0.767	0.779
				Upper bound	0.403	0.350	0.750	0.762
11	28	14	26	Baseline	0.347	0.283	0.614	0.621
				With Interaction	0.343	0.275	0.621	0.639
				Upper bound	0.361	0.293	0.643	0.646
12	39	22	49	Baseline	0.308	0.229	0.523	0.551
				With Interaction	0.304	0.221	0.523	0.551
				Upper bound	0.317	0.234	0.538	0.564

Table 6.4. IQE: Participants worked through fifty queries again. *# Expansion sub-sets selected* refers to the number of queries for which the participant chose an option. *Selections above baseline* refers to the number of times the option selected by the user was better than the baseline query. *Upper Bound* refers to the hypothetical situation where the user always selects the option that leads to the highest MAP, and chooses the baseline when none of the options are better than it. All scores, including upper bounds, were calculated only considering the queries for which the participant selected a sub-query.

3. *possibly. It can be helpful in identifying when and why a query is bad because the corpus contains something other than what I'm expecting.*

4. *No: I prefer a ranked list of unique documents spanned over several pages (or at least to a far rank depth).*

5. *yes and no, I like to see the article and the highlighted words that relates to the query but too many of the same article sometimes shows up. I would like one that shows itself once.*

The second, third and fifth comments are consistent with the goals of the user interface design, i.e. to provide users a preview of the different search spaces that their query can take them too, as well as the corpus contents. The first and fourth comments indicate that some users still prefer to use the popular ranked-list interface which gives them more power to re-specify their queries.

6.4 Conclusions

Our results clearly show there is much to be gained though IQR and IQE. While automatic techniques to perform query reduction and expansion are not successful, involving the user in the process clearly helped. We hypothesize that such interaction is useful for exploratory search where the user starts off with a more general information need and a looser notion of relevance. Successive rounds of interaction and query modifications are necessary to obtain the information desired. The interactive technique we have presented served as a bridge between the users and the IR system, helped adapt the users' queries to the characteristics of the retrieval algorithm and collection. By providing users a preview of the content retrieved by the options, a sense of the users' true information need was obtained to redirect the search towards relevant content.

6.5 Limitations to our study

The goal of the feasibility study was to determine if users could successfully select in a reasonable amount of time, a good option from a set presented to them. While we have shown that to be true, we acknowledge several limitations to our study. For instance, the users of our system had to work with prespecified queries, and not ones that reflected their personal information needs. This may have affected the participant's behavior and criteria used when assessing the options presented to them. More naturalistic settings would have provided greater credence to our conclusions. Our choice of queries, title and description portions of TREC topics, was motivated by the fact that these were standard test collections with relevance judgments readily available. However, these data sets were not developed with measurement of utility of user interaction in mind. We have also assumed that users will agree to enter long queries when asked to do so, and it is still an open question whether the long queries users enter will be similar in structure, form, and quality to the queries extracted from TREC topics. Further, since the topics are not categorized as fact finding, general, question-answering, transactional, navigational and so on extended analysis is not possible.

The demographics of our users included males and females between the age group twenty to thirty. This restricted profile could potentially affect the generalizability of our results and conclusions. The small number of users, twelve, further limits the strength of the conclusions. The results and conclusions are also particular to the interface we designed and presented: other interfaces could have led to worse or better results.

The user interaction paradigm we have explored involves a single round of interaction. In situations where users did not find any of the options presented to them useful, further rounds of interaction to reformulate or rephrase the query are called for. Exploration of session-based user interaction is planned for future work.

CHAPTER 7
SELECTIVE USER INTERACTION

We showed in the previous chapter that user interaction in the form of IQR and IQE for long queries can lead to significant improvements in performance over automatic systems. We consider user interaction as a resource to be judiciously used, preferably only when necessary. Invoking user interaction for every query is generally unwarranted - we show examples where interaction has potential to help only in a fraction of queries. Forcing the user to interact even when there is no *potential* for improvement can be detrimental to the user experience. We investigate IQR and IQE and develop procedures to detect the potential for improvement to help decide if user interaction is warranted. We show that by using these procedures we can avoid interaction for almost 40% of TREC queries without compromising significant improvements over the baseline. We also develop procedures to rank queries by their potential for improvement through user interaction. This enables systems to select the most promising queries for interaction when interacting with users working under time and cognitive load constraints.

7.1 Introduction

Our observations from the user study indicate that there were a large number of queries for which users did not select any of the options presented to them. This was because, frequently, none of the options (sub-queries or expansion subsets) selected by the automatic techniques and presented to the user were any better than the original query. In other words, the techniques to determine the correct options to present to the user are not perfect.

We believe that systems should be able to handle the situation discussed above, that invoking user interaction should be done in a judicious manner. Forcing the user to interact during every query session irrespective of whether there is utility in doing so can degrade overall user experience, and lead to increased cognitive load (Bruza et al., 1998).

Thus, there is clearly a need to develop a procedure to determine the utility of user interaction. Using such a procedure to determine beforehand the potential utility/futility of invoking user-interaction on a per-query basis will be useful in saving the user time and effort. Determining beforehand if a particular interaction had no potential would also provide a basis for attempting a different interaction mechanism, if available.

We investigated approaches towards determining when to interact with the user. We considered two settings - IQR and IQE. We based the decision to interact on the *potential* for improved performance with user involvement. Our experiments (Section 7.4) and results (Section 7.5) were simulations to enable abstracting away the effects of interface design, experimental methodology, subject experience and ability etc. We readily acknowledge that these factors might have important ramifications in a deployed system, but believe that their exploration is a natural extension for future work.

Our approach (Section 7.3) was to examine the properties of the set of options presented to the user. We hypothesized that useful sets of options will have distinguishing features from non-useful ones. We also believed that such a method will not be perfect, but users will be willing to trade a slight, but not significant drop in performance in return for a better search experience.

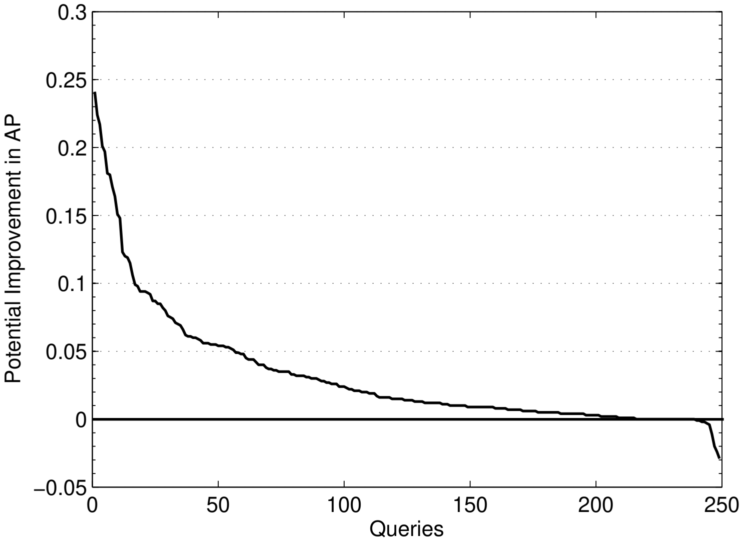


Figure 7.1. Query Reduction: The utility of interaction on a per-query basis. Values less than zero (to the right) indicate that none of the sub-queries presented to the user were better than the baseline query

7.2 User Interaction Potential

7.2.1 Query Reduction

Figure 7.1 sheds light on an interesting aspect of the potential improvements in performance that can be achieved with IQR. It shows the distribution of the *absolute* potential improvements in performance through user interaction across the 250 queries. If one were to consider a minimum improvement of 0.025 to be worth interacting to achieve, then we can see that user interaction for close to 150 queries is unnecessary. The overall improvements in MAP masked the minuscule improvements contributed by these queries.

Given this background, we seek to address the question, *Given a long query, is it possible to infer the potential utility of invoking user interaction to select a reduced version of the same query?*

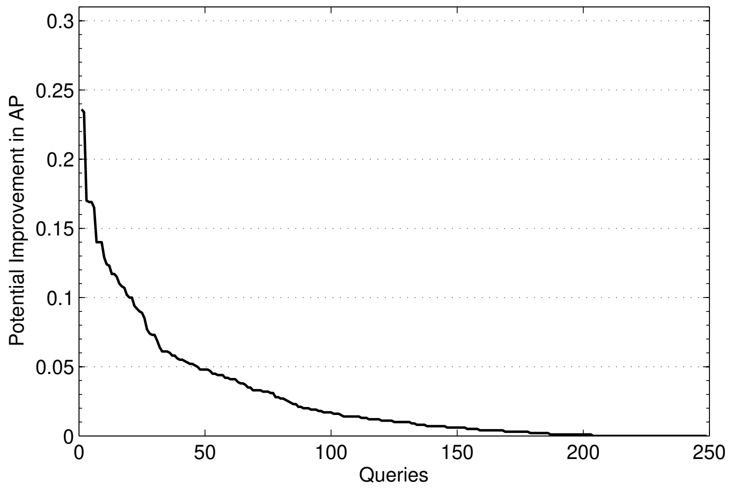


Figure 7.2. Query Expansion: The utility of interaction on a per-query basis.

7.2.2 Query Expansion

Figure 7.2 provides a pictorial description of the potential utility of user interaction with respect to IQE. Analogous to query reduction, we notice that user interaction for approximately 150 queries is of little utility.

Thus, we seek to address the question, *Given a short query, is it possible to infer the potential utility of invoking user interaction to select a better set of expansion terms?*

We believe that while there is demonstrable gain to be had from involving the user, it is equally important to determine when to bother a user with the request. In the following sections we will present techniques we developed towards this goal. We will present the results of simulated user studies.

7.3 Predicting Utility of Interaction

There is a large body of previous and related work on procedures to determine the quality of queries (Carmel et al., 2006; Cronen-Townsend et al., 2002; Zhou and Croft, 2006). The goal of that work was to predict in advance if a query will result in acceptable values of precision, and take appropriate steps if the query was predicted to fail (have a low AP). The procedures were thus tuned to accurately predict AP. Our goal is different. We wish to determine if the interaction techniques, IQR and IQE, will lead to an improvement in MAP. From the perspective of a user, expending interaction effort to improve precision from 0.1 to 0.11 is of the same utility as improving precision from 0.8 to 0.81 i.e. little utility. Hence we tuned our procedure to target absolute improvements in MAP, and not just MAP values themselves.

7.3.1 Qualities of good options

Our investigation of potential features for predicting improvement was guided by the following hypotheses about potentially good *options* for interaction. By options, we mean the set of top ten sub-queries or expansion subsets presented to the user.

1. When the original query is very long, a large number of extraneous terms are present that hinder retrieval instead of supporting it[1]. Thus, options that have low average length, or are derived from shorter queries, are potentially better

2. The average NE_MaxST scores of the set of options will be high, indicating a very focused set of queries

3. The scores of the sub-queries/expansion subsets in the options will be diverse, indicating that they cover different aspects of the query.

7.3.2 Query Reduction

For each query, we started with the top ten sub-queries ranked by NE_MaxST (Section 5.1.3). We used the scores assigned to them by the selection procedure to investigate several features based on measures of central tendency, measures of dispersion, and measures involving query lengths.

[1] Identifying and selectively weighting such terms is a continuing challenge.

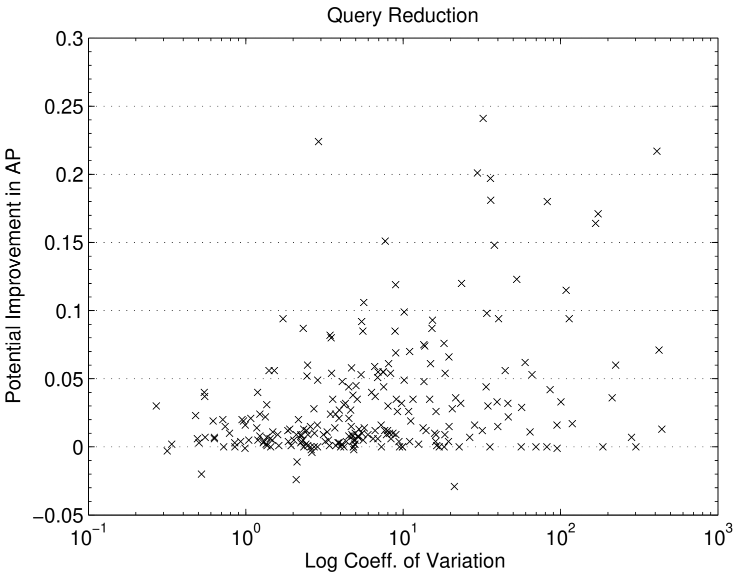


Figure 7.3. Scatter plot of log coefficient of variation versus potential improvement in AP due to user interaction

Table 7.1 provides a list of the features we experimented with, their correlation coefficients (γ) with potential improvements in AP, and the standardized regression coefficients obtained from multiple linear regression. While the correlation coefficients provided individual correlation of the features with potential improvement in AP, the standardized coefficients provided an overview of the relative importance of the features. One drawback of the standardized coefficients is that they do not reveal the dependence between variables (features). For example, Log Change Volatility (*lcv*), Percentage Change Volatility (*pcv*), and Coefficient of Variation (*cv*) features have an average correlation coefficient of 0.926 within themselves. Hence, we used both coefficients to choose the best features: standardized regression coefficients to reject some features and correlation coefficients to choose the best features. Features with standardized regression coefficients equal to zero were dropped from consideration.

The feature with the highest correlation coefficient was a dimensionless quantity, coefficient of variation (*cv*). *cv* can be considered as a measure of the scatter of a set

Feature	Formula	γ	Standardized Regression Coefficient
Original query length (ql)	-	-0.238	-0.000010
Mean score ($\bar{x}$)	$\bar{x} = \frac{1}{n}\sum_{i=0}^{n-1} x_i$	-0.226	0.000011
Geometric mean (gm)	$gm = (\prod_{i=0}^{n-1} x_i)^{\frac{1}{n}}$	-0.229	-0.000003
Harmonic mean (hm)	$hm = \frac{n}{\sum_{i=0}^{n-1}\frac{1}{x_i}}$	-0.232	0.000000
Interquartile mean (im)	$im = \frac{2}{n}\sum_{i=\frac{n}{4}}^{\frac{3n-4}{4}} x_i$	-0.227	0.000000
Root mean square (rms)	$rms = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} x_i^2}$	-0.223	-0.000004
Median	-	-0.226	0.000000
Variance (σ^2)	$\sigma^2 = \frac{\sum_{i=0}^{n}(x_i-\bar{x})^2}{n-1}$	0.084	0.000000
Standard Deviation (σ)	$\sigma = \frac{\sum_{i=0}^{n}(x_i-\bar{x})^2}{n-1}$	0.120	0.000000
Log Change Volatility (lcv)	σ of the natural base logarithms of the differences of successive ordered values of a set X $lcv = \sigma(Y)$; $y_i = \log\frac{x_{i+1}}{x_i}$	0.221	-3.141603
Percentage Change Volatility (pcv)	σ of the percentage differences of successive ordered values of a set X $pcv = \sigma(Y)$; $y_i = 100 * \frac{x_{i+1}-x_i}{x_i}$	0.263	0.000668
Coefficient of Variation (cv)	$cv = 100 * \frac{\sigma}{\bar{x}}$	0.314	0.000047

Table 7.1. Features with the highest correlation coefficient with respect to potential improvement in AP for IQR

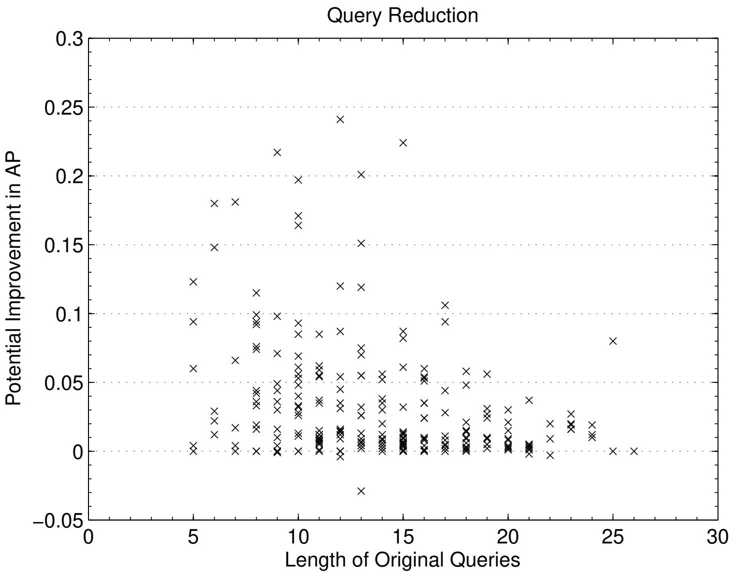


Figure 7.4. Scatter plot of original query size versus potential improvement in AP due to user interaction

of values. The positive correlation indicates that options that have high dispersal are more likely to contain sub-queries that lead to improvements in AP. This is consistent with our hypothesis that varied options were more likely to cover concepts the user is interested in.

The feature with the second highest correlation was Percentage Change Volatility (*pcv*). Since the correlation of *pcv* with *cv* was very high (0.934), we dropped that feature from consideration because of its redundancy with respect to *cv*. Further, the feature with the highest standardized regression coefficient, Log Change Volatility (*lcv*), too had a high correlation with *cv* (0.8874), leading to using only *cv* as a predictive feature. The original query length (*ql*) was the feature with next highest correlation with potential improvement in AP. It has a correlation coefficient of -0.512 with *cv*. The negative correlation with potential improvement in AP indicates that high values of initial query length translate to low-quality sub-queries, while lower values of initial query length are predictive of high-quality sub-queries. This is intuitive as identifying all the concepts in longer queries is more difficult. Longer queries also tend to induce more errors into the sub-query ranking procedure. Figure 7.4 is a scatter plot of original query size versus potential improvements in AP for the training queries. We can observe a gradual decrease in potential effectiveness as the length of the original query increases. He and Ounis (2004) too utilized query length as a feature in their attempts to predict query performance.

Figure 7.3 is a scatter plot of log of coefficient of variation versus potential improvements in AP for all the training queries. We can notice that higher improvements in AP are observed at the higher end of the *cv* scale.

7.3.3 Query Expansion

For each query we considered the top ten expanded queries ranked by NE_MaxST. Table 7.2 lists the features that correlated best with potential improvements in MAP.

cv was the feature that correlated most with the potential improvement in AP, while *pcv* came second. *lcv* and *pcv* had standardized regression coefficients of -587909.722 and 63.578 respectively. Since they both had very high correlation with *cv* (0.898 and 0.899 respectively), and to maintain consistency with the feature selected for IQR, we chose to use only *cv* as a predictive feature for IQE.

Figure 7.5 presents a scatter plot of *cv* versus potential improvement in AP. We notice trends similar to that observed in query reduction.

7.4 Usage Scenarios

We use predictive features in two different interaction scenarios. The first one is system-centric: i.e., the system learns and uses a technique to decide on user interaction each time a user issues a query. The second is user-centric: i.e., a user approaches a system with a set of queries, along with constraints on how much interactive effort she is willing to put in.

Feature	Formula	γ	Standardized Regression Coefficient
Mean score ($\bar{x}$)	$\bar{x} = \frac{1}{n}\sum_{i=0}^{n-1} x_i$	0.243	0.000000
Geometric mean (gm)	$gm = (\prod_{i=0}^{n-1} x_i)^{\frac{1}{n}}$	0.243	0.000000
Harmonic mean (hm)	$hm = \frac{n}{\sum_{i=0}^{n-1}\frac{1}{x_i}}$	0.241	0.000000
Interquartile mean (im)	$im = \frac{2}{n}\sum_{i=\frac{n}{4}}^{\frac{3n-4}{4}} x_i$	0.242	-80.396663
Root mean square (rms)	$rms = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} x_i^2}$	0.243	0.000000
Median	-	0.243	23.962995
Variance (σ^2)	$\sigma^2 = \frac{\sum_{i=0}^{n}(x_i-\bar{x})^2}{n-1}$	0.243	-442.25526
Standard Deviation (σ)	$\sigma = \frac{\sum_{i=0}^{n}(x_i-\bar{x})^2}{n-1}$	0.242	-44.740637
Log Change Volatility (lcv)	σ of the natural base logarithms of the differences of successive ordered values of a set X $lcv = \sigma(Y);\ y_i = \log\frac{x_{i+1}}{x_i}$	0.247	-587909.722
Percentage Change Volatility (pcv)	σ of the percentage differences of successive ordered values of a set X $pcv = \sigma(Y);\ y_i = 100 * \frac{x_{i+1}-x_i}{x_i}$	0.248	63.57823
Coefficient of Variation (cv)	$cv = 100 * \frac{\sigma}{\bar{x}}$	0.267	-0.010412

Table 7.2. Features with the highest correlation coefficient with respect to potential improvement in AP for IQE

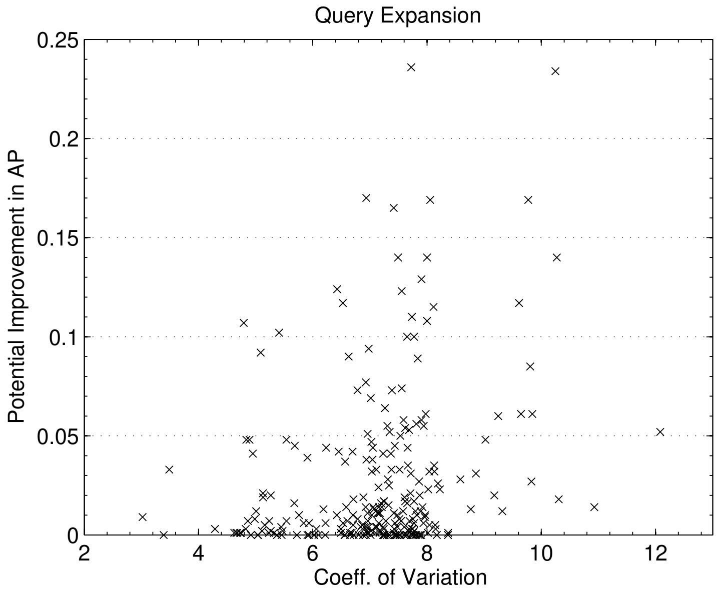


Figure 7.5. Scatter plot of coefficient of variation (*cv*) versus potential improvement in AP due to user interaction

7.4.1 System-Centric

Using training instances we learned a decision function to determine when to interact with a user. The high dispersion observed in Figures 7.4, 7.3, and 7.5 made the use of machine learning techniques like support vector machines (SVMs) to learn classifiers difficult. We observed that the classifier learned using SVMs used almost every training instance as a support vector - i.e., over fitting occurred. With this in view, we decided to apply thresholds to the feature values, and build a simple decision tree.

Query Reduction

Table 7.3 reports the change in potentially achievable MAP as well as the percentage of queries requiring user interaction when simultaneous threshold sweeps on both features, *ql* and *cv*, were performed. Every MAP value in the table is a statistically significant improvement over the baseline of 0.235. Statistical significance tests were performed using the paired t-test, with α set to 0.05.

		Coefficient of Variation Threshold							
		1	2	3	4	5	6	7	8
Query Length Threshold	15	0.261, 57	0.260, 56	0.259, 53	0.258, 50	0.258, 46	0.257, 43	0.256, 40	0.254, 37
	16	0.262, 64	$\boxed{0.262, 61}$	0.260, 57	0.259, 53	0.258, 49	0.257, 45	0.256, 42	0.254, 38
	17	0.263, 68	0.262, 65	0.260, 59	0.259, 55	0.259, 50	0.257, 45	0.256, 42	0.254, 38
	18	0.264, 73	0.263, 69	0.261, 63	0.260, 57	0.259, 50	0.257, 45	0.256, 42	0.255, 38
	19	0.265, 77	0.263, 71	0.261, 64	0.260, 58	0.259, 51	0.257, 45	0.256, 42	0.255, 38
	20	0.265, 79	0.264, 72	0.261, 64	0.260, 58	0.259, 51	0.257, 45	0.256, 42	0.255, 38

Table 7.3. Query Reduction: Effect on potential improvement in MAP due to simultaneously varying *ql* and *cv* thresholds. The numbers provided are <MAP,%queries requiring interaction> tuples. For example, to potentially achieve a MAP of 0.265 (last row, first column), we need to interact with the user for 79% of the test queries. The baseline was 0.235

Coefficient of Variation Threshold									
1	2	3	4	5	6	7	8	9	10
0.289, 100	0.289, 100	0.289, 100	0.289, 98	0.288, 93	0.286, 81	0.282, 61	*0.269*, 14	*0.266*, 6	*0.263*, 2

Table 7.4. Query Expansion: Effect on potential improvement in MAP at various *cv* thresholds. The numbers provided are <MAP,%queries requiring interaction> tuples. An italicized score implies that it was not a statistically significant improvement over the baseline MAP of 0.261

It is apparent from the table that a wide selection is available for determining appropriate thresholds for the two features. We chose values of 16 for ql, and 2 for cv (see box in Table 7.3). For the training set, it meant obviating interaction for 97 = ((1.0 – 0.61)*249) queries for a 2% reduction in potential MAP improvement . Function 7.4.1 presents the technique we adopted to determine the utility of interacting with the user in IQR.

Function 7.4.1: QRPREDICT(ql, cv, q_i)

```
$t_1 \leftarrow 16$
$t_2 \leftarrow 2$
if $ql[q_i] \leq t_1$ and $cv[q_i] \geq t_2$
  then $decision \leftarrow$ Interact
  else $decision \leftarrow$ Do not interact
return $(decision)$
```

Query Expansion

Table 7.4 reports the change in potentially achievable MAP and the number of queries requiring user interaction as a threshold-sweep is performed on cv. The transition to non-significant improvements over the baseline as the threshold is increased shows the limit to which we can *avoid* user interaction without impacting performance seriously.

Function 7.4.2 presents the technique we developed to determine the utility of interacting with the user in IQE.

Function 7.4.2: QEPREDICT(cv, q_i)

```
$t_1 \leftarrow 6$
**if** $cv[q_i] \geq t_1$
  **then** $decision \leftarrow$ Interact
  **else** $decision \leftarrow$ Do not interact
**return** $(decision)$
```

7.4.2 User-Centric

Consider the scenario where a user presents the system with a set of queries along with a condition that she is willing to only interact for $x\%$ of the queries. We imagine such a scenario could occur when a time-constrained user is performing exploratory search, for example searching for *vacations in Italy*, and hence would submit a series of queries to get all the information required. To maximize the benefit from user interaction, it is appropriate for the system to determine the $x\%$ of queries that would have most potential for improvement. We now present some experiments that use the *cv* values to make those decisions, and study its utility.

Query Reduction

We utilized the general trend observed in Figure 7.3 to guide the choice of queries to present for interaction. Higher values of potential improvements in AP generally imply higher values of *cv*. Our approach was to sort the options in the descending order of *cv* values, and present them to the user in that order.

Figure 7.6 shows the utility of the approach when the user accedes to interact for 10%, 20%, 30% and so on of the query set. The lowest curve shows the gradual improvements with increased user interaction when query subsets are chosen at random for interaction. The highest curve tracks the improvement when the system makes the best choice (highest potential improvement in MAP) on queries for interaction

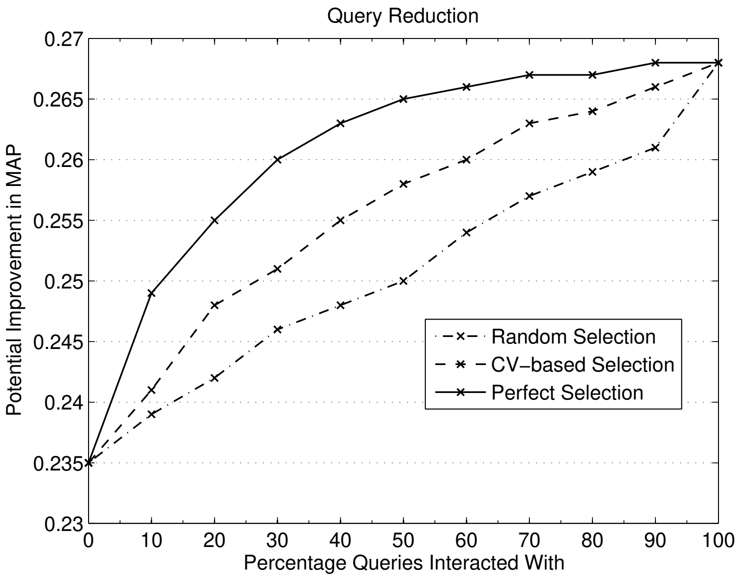


Figure 7.6. Query Reduction: Utility of various query ordering procedures when the user places constraints on the number of interactions

each time. In between the two is the curve that conveys the effect of presenting the options in descending order of *cv*. While the potential for improvement does not rise as rapidly as in the upper bound case, it clearly is much better than presenting the user with queries in random order, and definitely better than no interaction at all.

Query Expansion

We noticed trends similar to Figure 7.3 for query expansion too, and followed the exact same procedure we adopted in the query reduction case. Figure 7.7 depicts similar exploration of the utility of ranking the options by *cv* before presenting them to the user.

7.5 Results of Selective Interaction

In this section we present results of using our techniques on different sets of queries and collections in the context of IQR and IQE.

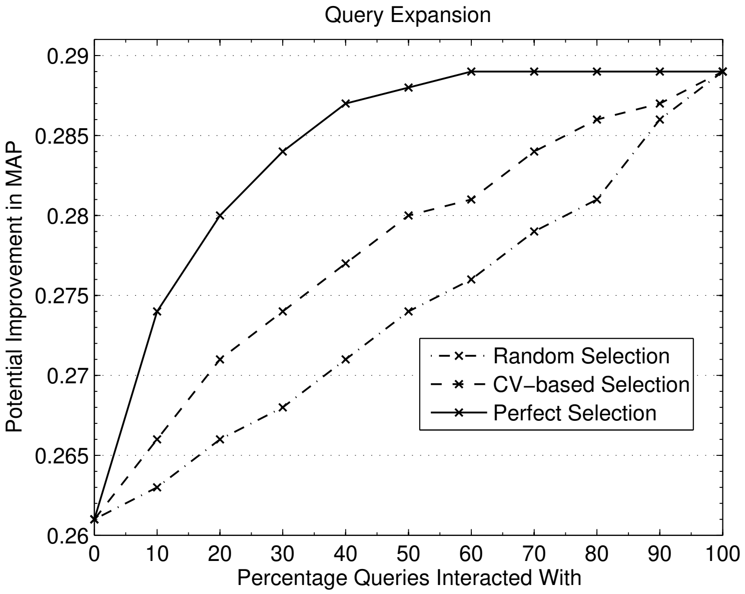


Figure 7.7. Query Expansion: Utility of various query ordering procedures when the user places constraints on the number of interactions

7.5.1 Query Reduction

In Table 7.5 we provide an overview of results when the system makes a decision to either interact with the user or go with the baseline query. We can see that when selective interaction was performed there was an average drop of 40% in the number of queries the user had to interact with, leading to an average drop in performance of 4.6%. In spite of the reduction, the final MAP was significantly better than the baseline (paired t-test text,α=0.05). However, in the case of Robust 2005 and HARD 2003, there was a significant drop in performance from what would have been achieved if the user interacted with all the queries ('User Select'). For a user with only enough time to interact for 60% (or *not* interact with 40%) of the queries the significant improvement over the baseline is still worth it.

Figure 7.8 provides an overview of the performance impact on Robust 2005 as the percentage of interactions is increased. The discrepancy in correspondence between the MAP at 60% interaction in the graph and the value reported in the table is because the latter's ordering of queries involves the second feature *ql* too. Using the *cv* values to rank queries for interaction is significant better at the lower end of the X scale than going with a random selection. For the same user with time to spare for 60% of the queries, we can observe that using *cv*-based selection helps obtain better performance with the *same effort*, when compared to randomly selecting queries.

7.5.2 Query Expansion

The results for our experiments with query expansion are given in Table 7.6. Again, we observed statistically significant improvements over the baseline for all three collections. The greatest reduction in the number of queries requiring interaction was for HARD 2003. As expected, the MAP achieved by our system was numerically less than that potentially achieved by interacting with all queries.

Figure 7.9 shows the potential gains obtained by increased user interaction on the Robust 2005 corpus. We notice that in the ideal case upper bound performance can be achievable by interacting with only 50% of the queries. In other words there was no utility in interaction for 50% of the queries. This explains the occasional 'flattening' of the *cv*-based selection and Random selection curves. The lower portion of the *cv*-based selection curve has a higher slope than the upper portion. This indicates that the selection process had done a good job of presenting queries with higher potential ahead of those with less.

7.5.3 Summary

For both query reduction and query expansion for all data sets we can notice small degradations in potential improvements in performance as we avoid interacting with the user for some (percentage of) queries. Thus, there exists a tradeoff that the designer of a system making use of our predictive techniques will need to consider. This tradeoff is between the amount of interaction and the degradation in potential

	Robust 2005	TREC 5	HARD 2003
Baseline	0.160	0.142	0.227
Upper Bound	0.283	0.217	0.351
Auto Select	0.162	0.122	0.223
User Select	0.190	0.158	0.267
Thresholded Select	0.180	0.153	0.253
% drop in MAP	5.5	3.1	5.2
% queries dropped	42	40	44

Table 7.5. Final MAP value results for query reduction.

	Robust 2005	TREC 5	HARD 2003
Baseline	0.239	0.159	0.315
Upper Bound	0.305	0.210	0.371
Auto Select	0.244	0.162	0.319
User Select	0.266	0.170	0.333
Thresholded Select	0.260	0.165	0.325
% drop in MAP	2.2	2.9	2.4
% queries dropped	22	32	52

Table 7.6. Final MAP value results for query expansion.

improvements in performance, something that the designer needs to tune for based on values of the latter that she considers acceptable.

7.6 Related Work

Shen and Zhai (2005) presented work that also dealt with the efficiency of user interaction. They performed simulated user studies for interaction involving document-level feedback, with the goal of developing procedures that chose the best documents from a pool to present to the user for feedback. The procedures they developed for and results from such *active feedback* showed that showing users a diverse set of documents was most effective. However unlike our work on query reformulation, they did not extend theirs to determine when to interact with the user, or how to handle a user with time and cognitive load constraints.

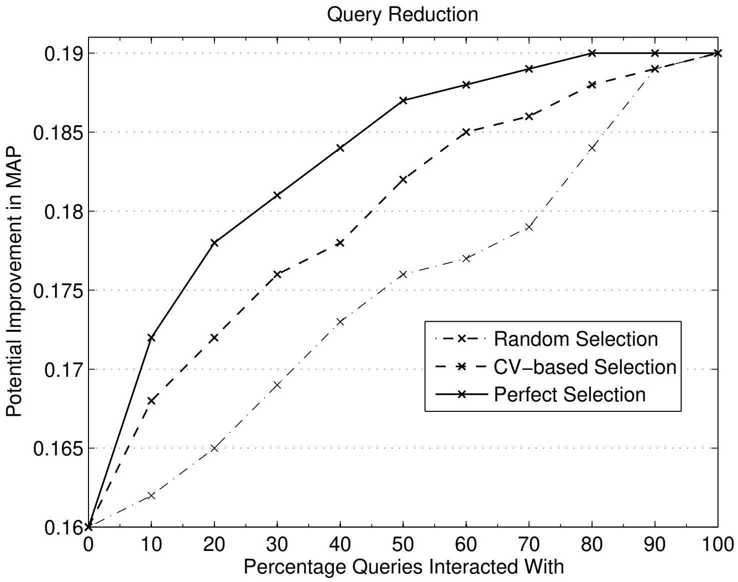


Figure 7.8. Trajectories of potential improvements in MAP using various question-selection techniques for Robust 2005 in IQR

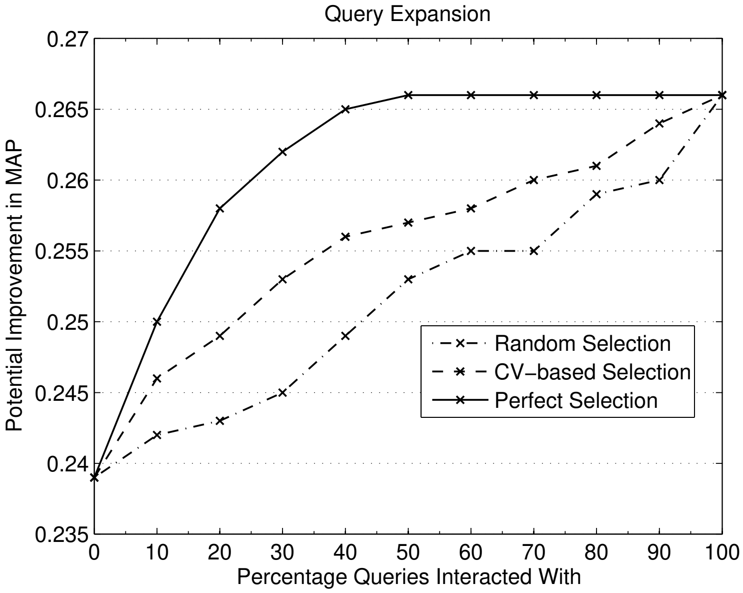


Figure 7.9. Trajectories of potential improvements in MAP using various question-selection techniques for Robust 2005 in IQE

7.7 Conclusions

We have discussed an important problem concerning adaptive information retrieval systems. While user interaction is a promising way to improve retrieval effectiveness, its efficiency needs to be considered too. Inefficient interactive systems that force a user to interact on every instance can cause disenchantment. We have shown that it is possible to predict the utility of interaction with reasonable accuracy, and use it without compromising much on effectiveness. The use of a single feature measuring scatter for both interaction mechanisms implies that interaction mechanisms that provide a wide range of choices have more utility. In other words, showing the user the different parts of the search space her query could lead her to is advantageous.

CHAPTER 8
EFFICIENCY

We have shown in previous chapters that richer expressions of information need by users can be leveraged to improve search performance. While the richer expressions we dealt with are of a particular type namely TREC *description* queries and expanded *title* queries, other expressions can take the form of inclusion of additional terms the user believes are related to the query (Kelly et al., 2005), identifying documents containing similar information (Smucker and Allan, 2006), identifying topics the query is related to (Kumaran and Allan, 2006a) and so on. We continue our focus on interactive handling of long queries through IQR and IQE.

The IQR and IQE techniques we have developed involve asking a user to select from ten options for each and every query. This can be detrimental to the user experience as user interaction requires cognitive and physical effort from the user. Developing techniques to identify and present a minimal set of options to users is thus important. One way to identify a minimal set of options is to identify redundant options and drop them. In the context of information retrieval, redundant options are those that lead the user to the same search space. We demonstrate the similarity of the problem of identifying redundant options with *Set Cover*, an NP-Complete optimization problem. We utilize a known greedy algorithm to provide an approximate solution (Section 8.2), and show that we can successfully remove redundant options and retain useful ones..

Generating options for the IQR and IQE interfaces involved analyzing all possible combinations of the terms in the long query (*sub-queries*) or set of terms suggested by PRF (*expansion subsets*) to determine the set of top *options* to present to users. Such a technique is difficult to realize in practice due to the exponential number of options that need to be analyzed. In Section 8.3 we present a technique based on analyzing the properties of *ideal* queries, and using those observations to prune the option search space.

Since we are improving on the interaction techniques presented in the previous chapter, in this chapter we will confine ourselves to reporting performance on simulated user studies. We hypothesize that improvements such as more efficient background processes, fewer options presented to users, and better quality of options will naturally extend to improving the interaction experience and performance.

8.1 Motivation

To identify top-ranking options we represented each of the 2^n options as a graph constructed with the constituent terms as vertices, and the mutual information (Church and Hanks, 1990) between the terms as edge weights. n is the length of the original TREC description query in the case of IQR, and the number of expansion terms in IQE. Following Rijsbergen (1979) the maximum spanning tree (Cormen et al., 2001) was identified on each graph, and its weight used to represent the quality of the option. After ranking the entire set of options by the weight of their corresponding maximum spanning trees, the top ten were selected.

Table 8.1 shows the best performance that can be achieved under various conditions. All results are reported for 249 TREC *description* queries from the Robust 2004 track. We will treat the description portion of TREC queries as long queries for our experiments. Baseline refers to a query-likelihood (QL) run using the Indri search engine (Strohman et al., 2005), while PRF refers to automatic query expansion using PRF. "Interaction Upper Bound" refers to the upper bound on the performance that can be obtained from user interaction, i.e. the user always selects the best option

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.472	0.397	0.379	0.240
PRF (Best)	*0.514*	*0.442*	*0.423*	*0.288*
Query Reduction				
Interaction Upper Bound	**0.634**	**0.528**	0.498	*0.300*
Query Expansion				
Interaction Upper Bound	**0.571**	**0.480**	0.447	*0.292*

Table 8.1. The utility of IQR and IQE. Italicized values indicate that the scores are significantly better than the baseline, while those in bold are significantly better than PRF. Statistical significance was measured using a paired t-test, with α set to 0.05

from the ten presented. This is the baseline for the experiments we conduct in this chapter. It represents the situation where an exponential search has been performed to identify the top ten options to show to a user, and the user selects one from the top ten shown.

As mentioned before, the current interaction paradigm involves always presenting users with ten options for all queries. There is clearly scope for reducing the number of options presented to users, especially when on average only three out of ten of them are better than the baseline. Figure 8.1 is a histogram of the number of options better than the baseline for each of the 249 queries we used for training. Clearly, a large fraction of the options presented to users have no utility, and can potentially degrade the user experience. In Section 8.3 we present techniques that enable us to reduce the number of options we present to users significantly, without degrading performance.

8.2 Minimal Option Sets

The technique to analyze the options makes use of co-occurrence information of the constituent terms. While this provides a good sense of the *cohesiveness* of the option, it does not inform the user of the relative utility of an option with respect to the other ones shown. Given that some options differ by just a single term, it is quite likely that they might all direct the user to the same search space. In such cases it is

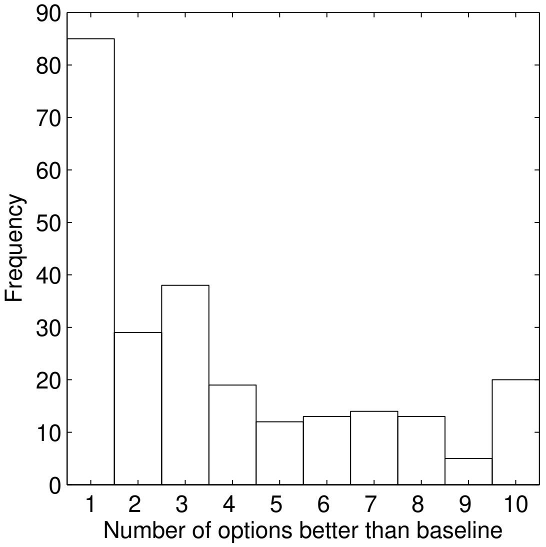


Figure 8.1. Distribution of the number of options better than the baseline query for a set of 249 training queries

wasteful to show similar options, and instead displaying a minimal subset of options that *covers* the original search space(s) might be better. This intuition forms the basis for our technique to prune the original set of options shown to the user. We introduce the *Set Cover* problem before going into the details of our technique.

The set covering problem (Cormen et al., 2001) is a NP-complete optimization problem. An instance $(X, \mathcal{F})$ of the set covering problem consists of a finite set X and a family $\mathcal{F}$ of subsets of X, such that every element of X belongs to at least one subset in $\mathcal{F}$. Mathematically,

$$X = \bigcup_{S \in \mathcal{F}} S \tag{8.1}$$

The family F is said to *cover* the elements in X. The goal is to find a minimum-size subset $\mathcal{C}$, $\mathcal{C} \subseteq \mathcal{F}$, whose members cover all of X i.e.

$$X = \bigcup_{S \in \mathcal{C}} S \tag{8.2}$$

Since finding the exact solution is NP-Complete, we used a greedy set cover algorithm (Cormen et al., 2001) that works by selecting the subset that covers the most elements in X in an iterative fashion. The advantage of using the greedy algorithm is that it not only runs in time polynomial in $|X|$ and $|\mathcal{F}|$ but also returns a $(\ln(|X|)+1)$ approximation to the solution.

The greedy approximation algorithm for the set cover problem is provided below.

```
GREEDY-SET-COVER($X, \mathcal{F}$)
$U \leftarrow X$
$\mathcal{C} \leftarrow \emptyset$
while $U \neq \emptyset$
    do select an $S \in \mathcal{F}$ that maximizes $|S \cap U|$
        $U \leftarrow U - S$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{S\}$
return $\mathcal{C}$
```

8.2.1 Overlapping Search Results

We now show how the problem of finding minimal option sets can be cast as a set cover problem. Each option is used as a query to retrieve a set of ten documents. Let X be the union of sets of documents retrieved. The sets of documents correspond to the family $\mathcal{F}$ of subsets whose union is X. Our goal is to identify a minimal set of subsets $\mathcal{C}$ from $\mathcal{F}$ that cover X.

Table 8.2 shows the impact on performance metrics as well as the number of options presented to users due to the various techniques. We can observe in the case of “Set-Cover-based Pruning” for IQR that for an average decrease of two options per query, there is no (statistically) significant drop in performance. In the case of IQE, it is drastic: an average reduction of six options per query without significant

System	P@5	P@10	NDCG@15	MAP	Avg. # options
Baseline (QL)	0.472	0.397	0.379	0.240	-
Interactive Query Reduction (IQR)					
Interaction Upper Bound	0.634	0.528	0.498	0.300	9.7
Set Cover-based Pruning	0.619	0.523	0.488	0.293	7.5
Snippet-based Pruning	0.610	0.510	0.483	0.290	6.7
Interactive Query Expansion (IQE)					
Interaction Upper Bound	0.571	0.480	0.435	0.292	10.0
Set Cover-based Pruning	0.560	0.479	0.437	0.294	3.8
Snippet-based Pruning	0.535	0.454	0.421	0.285	2.1

Table 8.2. Effect of option-pruning strategies on performance metrics, and number of options presented to users

performance loss. This result for IQE can be understood by considering the fact that query expansion results in a much longer query than the original, and the subtle differences between options (usually by a term or two) do not lead to radically different sets of documents being retrieved. In summary, judging by the insignificant drops in performance, we successfully retained the useful options and removed the redundant ones.

8.2.2 Identical Snippets

The user is guided in making a decision on which option to select using a snippet of text. This snippet is extracted from the top-ranking document that is retrieved when the option is used as a query. Frequently, the snippets returned by different options are the same, making the task of selecting an option difficult. Retaining a single option from the set that retrieves the same snippet can further decrease the number of options presented to users. The results for “Snippet-based Pruning” in Table 8.2 show the impact of this pruning strategy. With the exception of P@5 for IQE, this strategy results in an average reduction of approximately one option without significantly impacting performance.

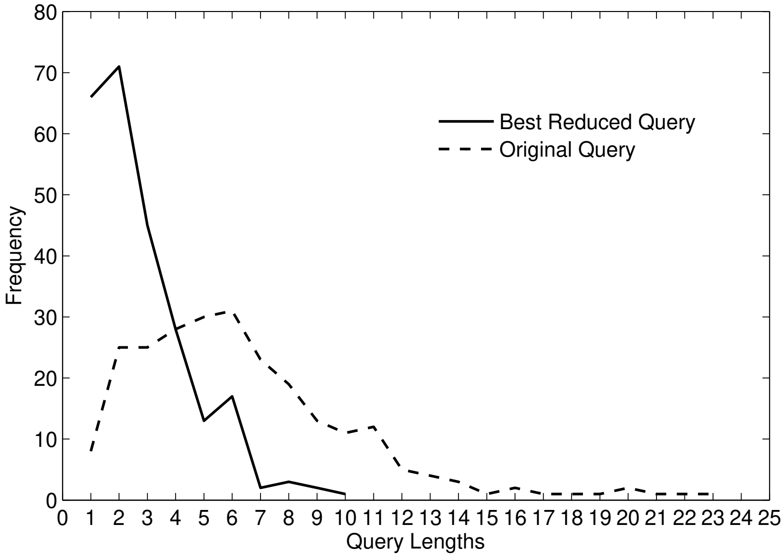


Figure 8.2. Distribution of lengths of original and best reduced queries

8.3 Option Analysis

A good query is long enough to describe key concepts but also short enough to avoid containing unnecessary terms. To determine the appropriate length of sub-queries, we plotted the distribution of the query lengths of the best performing sub-query for each query in our training set. Figure 8.2 shows the distribution, and compares it with the distribution of the lengths of the original queries. We can observe that the best sub-queries are never more than ten terms in length, with most having six or fewer. This observation informed the decision to restrict analysis of sub-queries to those of length less than or equal to six. Table 8.3 shows the impact on "Interaction Upper Bound" performance due to this restriction. We notice that the restriction not only results in a reduced number of sub-queries analyzed (Figure 8.3) but also maintains the potential for improvement through IQR.

In a similar fashion, we analyzed the size of the best expansion subset for our training queries (Figure 8.4). We can observe that the best expansion subsets are frequently between eight to twelve terms in length. This observation again informed

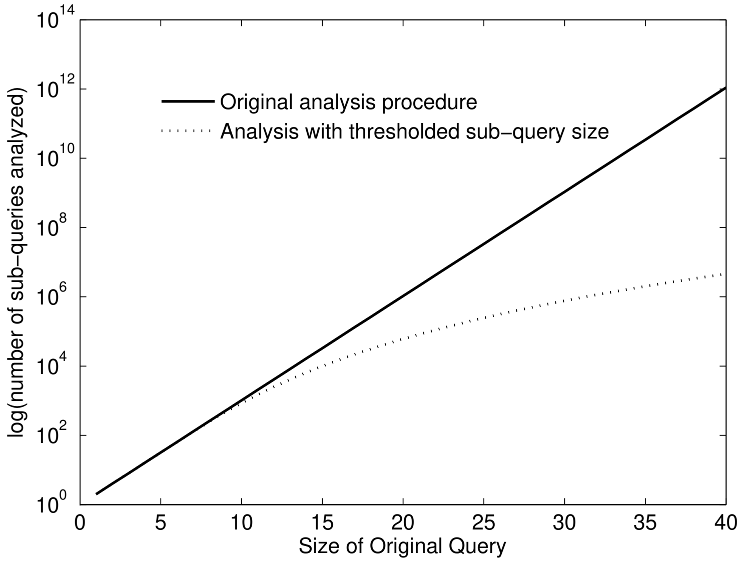


Figure 8.3. Growth of number of sub-queries analyzed with original query size

System	P@5	P@10	NDCG@15	MAP
Interactive Query Reduction (IQR)				
Interaction UB	0.634	0.528	0.498	0.300
Interaction UB with Size Threshold	0.634	0.528	0.498	0.300
Interactive Query Expansion (IQE)				
Interaction UB	0.571	0.480	0.447	0.292
Interaction UB with Size Threshold	0.578	0.485	0.450	0.303

Table 8.3. Effect of thresholding the lengths of options analyzed. UB refers to Upper Bound.

% Reduction in MAP	Average Size of Expansion Subset	MAP
1	8.57	0.367
2	7.99	0.365
3	7.63	0.363
4	7.41	0.361
5	7.26	0.360
6	7.15	0.359
7	7.06	0.358
8	7.02	0.358
9	6.95	0.357
10	6.89	0.357

Table 8.4. The tradeoff between reduction in achievable MAP and the average length of expansion subsets.

the decision to restrict analysis of expansion subsets to those of length less than or equal to twelve. The section for IQE in Table 8.3 conveys that this restriction actually helped avoid some bad options, and raised the potential for improvement through IQE.

While we are considering the size of the best performing IQE queries, there is also a possibility that there are queries exhibiting similar performance, but of shorter length. The distribution of query lengths when we consider the shortest of IQE options that perform just 1% worse then the best is included in Figure 8.4. We can observe that the distribution shifts to the left indicating that we can restrict analysis to smaller sized subsets if we are willing to accept a 1% drop in potential MAP improvement. Agreeing to take a 10% drop results in further reduction of the subset sizes we need to consider. Table 8.4 summarizes the tradeoff between the percentage drop in potentially achievable MAP through IQE and the average size of subsets that need to be analyzed. The table shows the possibility of further reduction in the number of options analyzed, but with a corresponding reduction in the potentially achievable MAP though user interaction.

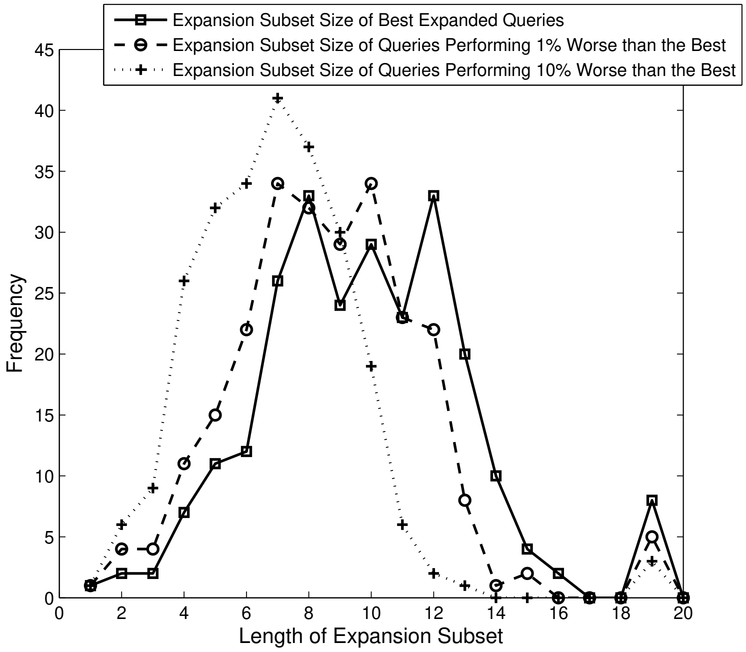


Figure 8.4. Length distribution of best expansion subsets

8.4 Conclusions

We have presented techniques to improve the efficiency of user interaction for information retrieval using long queries. Presenting users with the right number of options is an often-ignored aspect of interactive information retrieval. We have developed an approach for identifying a minimal set of options, and demonstrated that this technique retains good options and removes the redundant ones. In the case of IQE, we could drop 80% of the options without a decrease in performance, i.e. we can get the same performance by showing the user just two options. We hypothesize that this pruning technique, which can be adapted for use with any interaction technique, will enhance effectiveness by reducing the cognitive load on users. The exponential-sized analysis of options has been shown to be unnecessary, and reduced to polynomial-sized analysis without degrading performance. The ease with which the analysis process can be parallelized (different machines can analyze options of different lengths and follow up with a merge) and the reduction in complexity can pave the way for live deployment of the effective interaction techniques we have presented in this thesis.

CHAPTER 9

SELECTIVE INTERACTIVE REDUCTION AND EXPANSION

Our techniques to handle long queries involves either pruning the query to retain only the important terms (reduction), or expanding the query to incorporate related concepts (expansion). We have shown that roughly 25% performance improvements in terms of MAP can be realized through IQR and IQE alone. Most research in interactive information retrieval has over the years adopted a one approach-fits all strategy. In other words, irrespective of the query, the interaction is based on a fixed strategy. Since it is well known that queries fail for different reasons (Harman and Buckley, 2004a), it is important to explore query-specific interaction. This is particularly useful when the goal is to minimize user interaction. Asking users to answer only questions that have the most potential for improving the retrieval process can lead to better user experience. Guided by the philosophy that users must get the maximum benefit (effectiveness) for their investment of time and effort, we explore ways to make user interaction for long queries more effective in this chapter. We will show that selectively reducing or expanding a long query leads to an average improvement of 51% in MAP over the baseline for standard TREC test collections, and demonstrate how user interaction can be used to achieve this improvement. We finally present an analysis of long queries that continue to exhibit poor performance in spite of the improved techniques.

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.472	0.397	0.379	0.240
PRF (Best)	*0.514*	*0.442*	*0.423*	*0.288*
Query Reduction				
Upper Bound (UB)	**0.799**	**0.671**	0.626	**0.366**
Interaction UB	**0.634**	**0.528**	0.498	*0.300*
Query Expansion				
Upper Bound (UB)	**0.738**	**0.643**	0.587	**0.368**
Interaction UB	**0.571**	**0.480**	0.447	*0.292*

Table 9.1. The utility of IQR and IQE. Italicized values indicate that the scores are significantly better than the baseline, while those in bold are significantly better than PRF. Statistical significance was measured using a paired t-test, with α set to 0.05

9.1 Motivation

Table 9.1 (a reproduction of Table 8.1) shows the best performance that can be achieved under various conditions for 249 TREC *description* queries from the Robust 2004 track.

The results in Table 9.1 for query reduction and expansion show that user interaction can lead to significant improvements in performance for long queries. Further improvements can be obtained if we selectively invoke IQR or IQE. Figure 9.1 shows the ordered distribution of the difference between the potential gains due to IQR and IQE. Some queries are better suited for IQR, while others can be better improved through IQE. If we can selectively invoke IQR or IQE for each query we can potentially obtain a 51% (from 0.240 to 0.363, compared to 0.300 and 0.292 for only IQR and only IQE respectively) improvement in MAP over the baseline. Determining when to reduce and when to expand is similar in flavor to the problems of determining when to perform PRF (Cronen-Townsend et al., 2004) or when to perform stemming (Harman, 1991): correct answers to either can lead to significant improvements in performance. The tremendous scope for improvement makes the reduce/expand problem worthy of further investigation. In the next section we present a technique referred to as

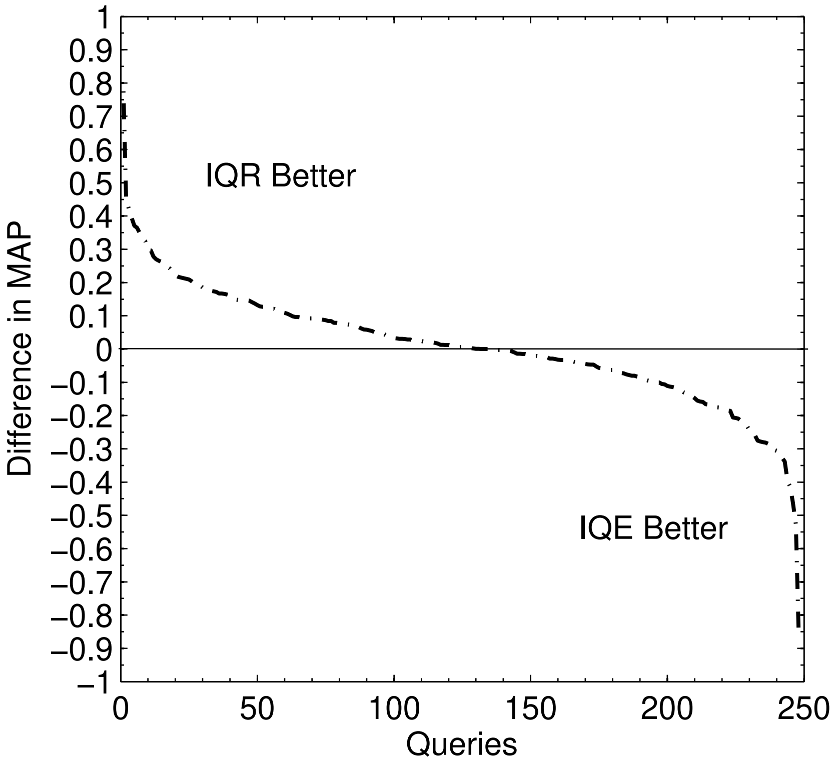


Figure 9.1. Difference in MAP due to Selective IQR and IQE

Selective Interactive Reduction and Expansion (SIRE), a hybrid of the IQR and IQE techniques.

9.2 Selective Interactive Reduction and Expansion (SIRE)

Tremendous gains in performance can be obtained by selectively expanding or reducing long queries. For each long query, our approach involved selecting the top five sub-queries and top five expansion sets and providing the user a merged list for interaction. The downside of this technique was that we risked losing potentially useful options ranked between six and ten. However, as Table 9.2 shows, this risk was insignificant when compared to the potential for improvement through SIRE. By viewing this mix of expansion and reduction options, along with a snippet of text to guide selection, the user can implicitly guide the system towards expansion or reduction of the query.

Table 9.2 summarizes the improvements in performance that can achieved using the SIRE technique. Consider the "Best Performance" section of the table, which describes the situation when the best option from among the ones presented is always selected. When IQR and IQE are used with five options, the performance is as detailed. However, when the options are combined (SIRE$_{comb}$), the user can potentially achieve better performance than could be achieved with either IQR or IQE with not only five, but also ten options. Another interesting aspect of the result is that MAP is significantly improved. IQR is primarily a precision enhancing technique, while IQE is both precision as well as recall enhancing. The advantages of each technique have thus been carried over to the hybrid SIRE$_{comb}$ technique in the form of improved MAP. "Average Performance" reflects the situation when the user randomly selects an option from the top ten. We can see that randomly choosing from SIRE$_{comb}$ options is significantly better than choosing from those of IQR$_5$, IQE$_5$, and IQR$_{10}$. The only exception is IQE$_{10}$, but as we will show in Figure 9.2 SIRE$_{comb}$ improves performance for more queries than IQE$_{10}$.

9.3 Efficient SIRE

In Chapter 8 we presented strategies to prune the number of options presented to users for interaction. Table 9.3 shows the effects of applying the Set Cover and Snippet-based pruning strategies to the SIRE technique. Consider the section titled "Best Performance". In this hypothetical case which provides an upper bound on the performance achievable through user interaction, the results show that the pruning strategy works, and performance comparable to IQR$_{10}$ is attained by showing 50% less options. In the situation when the user selects randomly from the options presented i.e. "Average Performance", pruning results in a reduction in potential improvement

System	P@5	P@10	NDCG@15	MAP	Avg. num. options
Baseline	0.472	0.397	0.379	0.240	-
Best Performance					
IQR$_5$	0.554	0.469	0.448	0.274	4.9
IQE$_5$	0.555	0.466	0.435	0.292	5
SIRE$_{comb}$	0.654	0.552	0.521	0.347	9.9
IQR$_{10}$	0.634	0.528	0.498	0.300	9.7
IQE$_{10}$	0.571	0.480	0.450	0.292	10.0
Average Performance					
IQR$_5$	0.305	0.279	0.296	0.18	4.9
IQE$_5$	0.448	0.386	0.361	0.246	5
SIRE$_{comb}$	0.366	0.327	0.347	0.228	9.9
IQR$_{10}$	0.323	0.290	0.301	0.189	9.7
IQE$_{10}$	0.426	0.377	0.398	0.269	10.0
Worst Performance					
IQR$_5$	0.145	0.133	0.138	0.096	4.9
IQE$_5$	0.427	0.376	0.398	0.269	5
SIRE$_{comb}$	0.129	0.118	0.121	0.080	9.9
IQR$_{10}$	0.122	0.118	0.122	0.088	9.7
IQE$_{10}$	0.425	0.366	0.348	0.235	10.0

Table 9.2. The subscript in the system name indicates the number of options presented to users. The SIRE$_{comb}$ technique involves merging IQR$_5$ and IQE$_5$. For comparison, performance of IQR and IQE with ten options is also provided

System	P@5	P@10	NDCG@15	MAP	Avg. # options
Baseline (QL)	0.472	0.397	0.379	0.240	-
Selective Interactive Reduction and Expansion (SIRE)					
Best Performance					
Interaction Upper Bound	0.654	0.552	0.521	0.347	9.9
Set Cover-based Pruning	0.643	0.544	0.512	0.340	6.7
Snippet-based Pruning	0.629	0.53	0.505	0.335	5.5
Average Performance					
Interaction Upper Bound	0.366	0.327	0.347	0.228	9.9
Set Cover-based Pruning	0.333	0.301	0.320	0.206	6.7
Snippet-based Pruning	0.321	0.290	0.307	0.199	5.5
Worst Performance					
Interaction Upper Bound	0.129	0.118	0.121	0.080	9.9
Set Cover-based Pruning	0.137	0.121	0.124	0.081	6.7
Snippet-based Pruning	0.139	0.124	0.127	0.082	5.5

Table 9.3. Effect of option-pruning strategies on performance metrics, and number of options presented to users. “Best Performance” refers to the situation when the user always selects the best option from those presented to her. “Average Performance” refers to the situation when the user selects options randomly

with a corresponding decrease in number of options presented. Finally, in the “Worst Performance” situation where the user always selects the worst option, performance increases as pruning is applied to the options. This means that the pruning strategies discarded lower-quality options. Since “Average Performance” decreases while “Worst Performance” increases, it means that a mix of good as well as bad options are being discarded by the pruning strategies. However, the higher quality options are retained as is evident from the “Best Performance” numbers.

9.4 Results and Analysis

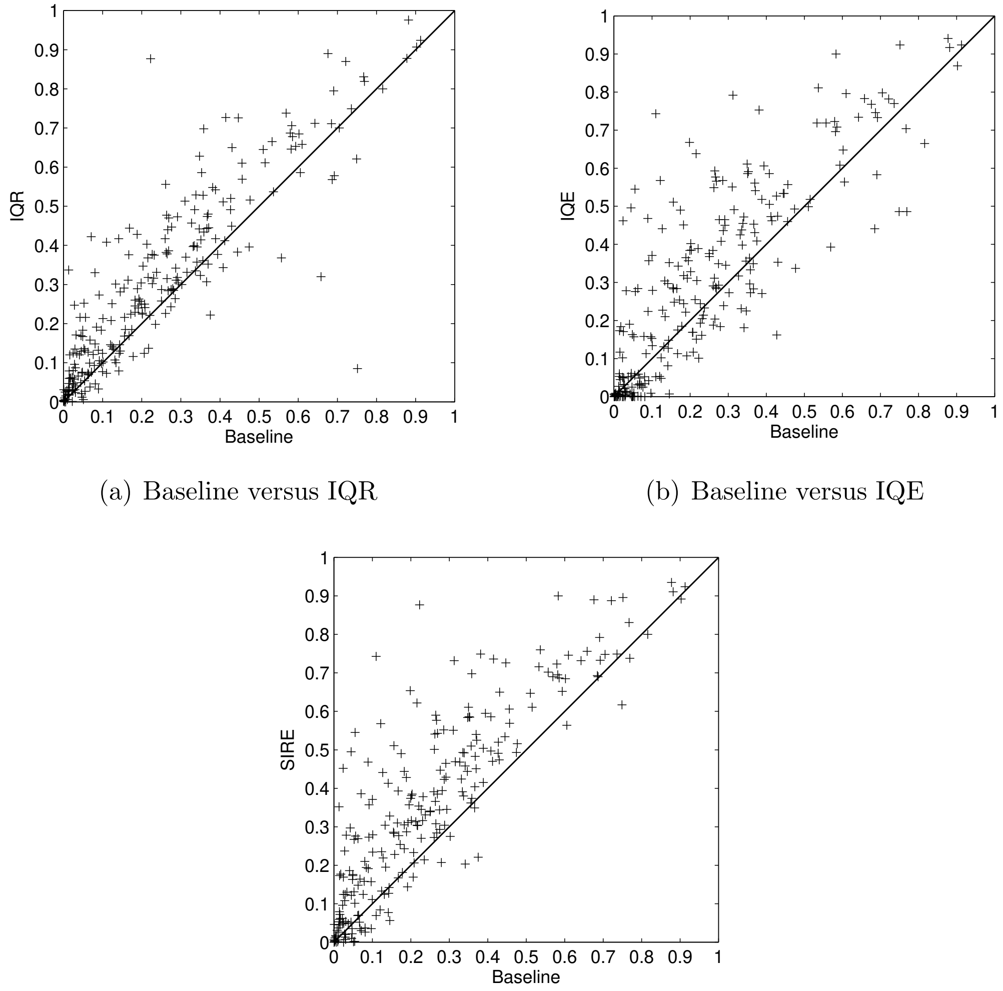
In this section we analyze the effect of using our techniques on different data sets, the results of which are presented in Table 9.4.

For all collections we notice trends similar to that observed for the Robust 2004 data set namely the higher performance of the SIRE system over IQR and IQE with not only five options but also ten. SIRE remains competitive or better even after

Corpus	System	P@5	P@10	NDCG@15	MAP	Avg. # options
Robust 05	Baseline	0.412	0.386	0.270	0.156	-
	IQR with 5 options	0.488	0.450	0.333	0.194	5.0
	IQE with 5 options	0.512	0.490	0.351	0.229	5.0
	SIRE with combined options	0.612	0.564	0.417	0.262	10.0
	SIRE with set cover-based pruning	0.604	0.556	0.415	0.257	7.2
	SIRE with snippet-based pruning	0.500	0.484	0.405	0.221	2.5
	IQR with 10 options	0.572	0.510	0.396	0.218	10.0
	IQE with 10 options	0.540	0.510	0.364	0.237	10.0
HARD 2003	Baseline	0.544	0.478	0.441	0.228	-
	IQR with 5 options	0.644	0.57	0.515	0.283	4.9
	IQE with 5 options	0.628	0.544	0.476	0.298	5.0
	SIRE with combined options	0.720	0.646	0.572	0.346	9.9
	SIRE with set cover-based pruning	0.720	0.636	0.570	0.343	6.8
	SIRE with snippet-based pruning	0.632	0.544	0.568	0.300	2.5
	IQR with 10 options	0.712	0.618	0.560	0.301	9.8
	IQE with 10 options	0.636	0.580	0.496	0.305	10.0
TREC 5	Baseline	0.384	0.322	0.305	0.163	-
	IQR with 5 options	0.372	0.308	0.327	0.154	4.9
	IQE with 5 options	0.352	0.302	0.312	0.166	5.0
	SIRE with combined options	0.444	0.370	0.389	0.202	9.9
	SIRE with set cover-based pruning	0.440	0.358	0.382	0.19	6.9
	SIRE with snippet-based pruning	0.348	0.298	0.378	0.161	2.3
	IQR with 10 options	0.468	0.374	0.386	0.171	9.7
	IQE with 10 options	0.368	0.312	0.320	0.168	10.0

Table 9.4. Summary of the results of using the SIRE and option-pruning techniques on test data sets

option pruning: with an average of six options it meets or beats 10-option IQR and IQE.



(a) Baseline versus IQR

(b) Baseline versus IQE

(c) Baseline versus SIRE

Figure 9.2. Scatter plots of baseline performance (MAP) and performance due to IQR, IQE, and SIRE. IQR and IQE used ten options, while SIRE used a combination of five options from each of IQR and IQE

We now analyze the performance of IQR, IQE, and SIRE with respect to the baseline (automatic) system. Figure 9.2 shows the scatter plots of the MAP values of the baseline system with respect to each of the interactive techniques, for 249 Robust 04 queries. The line $y = x$ is included to identify the queries that were improved or hurt by each technique. A point above the $y = x$ line means that performance was improved through interaction, while a point below the line means that interactive retrieval hurt the query. We can observe that a larger fraction of queries was improved by IQR (Figure 9.2(a)) in comparison to IQE (Figure 9.2(b)). The plot for IQE (Figure 9.2(b)) has greater spread, and higher density in the upper left hand corner compared to IQR. This means that when IQE helps, it helps to a greater extent than IQR. However overall improvements are mitigated by the fact that IQE performs worse on already poorly performing queries. The SIRE system combines the best of IQR and IQE. Not only are there fewer queries below the $y = x$ line, but the density in the upper left hand corner is greater. These observations mean that SIRE provides a more comprehensive improvement over a set of queries.

We now turn our attention to the lower left hand corner of Figure 9.2(c) - the area containing the set of queries that were not only poorly-performing to start with, but also were unaffected by IQR, IQE, and SIRE. We define poorly-performing queries as those that had a baseline, IQR, IQE, *and* SIRE MAP of less than or equal to 0.1. We analyzed each of the 45 such queries, and also the corresponding best reduced and expanded versions that were used to generate the "Upper Bound" scores in Table 9.1. For situations when the best reduced and expanded queries were themselves low-performing, it was clear that the user would have to enter a completely new query. Table 9.5 summarizes the failure categories we identified and suggests directions for future work. "System failure in identifying sub-query" (or "expansion set") refers to the situation when a better option was available, but the technique we used to rank the options failed to place it in the top 10. Of these, 3 of the options were of the type that a user with a similar information need could be expected to issue. Another 14 (10+4) of them would have been difficult for a human to come up with without a complete understanding of how the underlying search engine works. For the 4 queries for which Natural Language Processing (NLP) techniques would have worked, we expect that identifying noun phrases in the original query would have helped.

Analysis	Number of Queries	Example Query	Best Sub-query/ Expansion Subset
Term mismatch: new query required	24	Identify instances in which weather was a main or contributing factor in the loss of a ship at sea.	
System failure in identifying sub-query			
• Best sub-query incomprehensible to human	10	Identify a country or a city where there is evidence of human slavery being practiced in the eighties or nineties.	country city evidence slavery
• Human could have identified it	3	Isolate instances of fraud or embezzlement in the international art trade.	fraud art
• NLP techniques could have helped	4	Fiber optic link around the globe (Flag) will be the world's longest undersea fiber optic cable. Who's involved and how extensive is the technology on this system. What problems exist?	link globe flag *world's longest undersea fiber* cable involve technology
System failure in identifying expansion set	4		

Table 9.5. Breakdown of the analysis of low-performing queries. By NLP techniques, we refer to identification of phrases in the query and treating them as a unit.

9.5 Conclusions

We have presented techniques to improve the effectiveness and efficiency of user interaction for information retrieval using long queries. The SIRE technique, a hybrid of IQR and IQE, has been shown to be an extremely effective way to capitalize on the strengths of the two. By showing an optimal number of a mix of IQR and IQE options we can implicitly ask the user to decide whether query reduction or query expansion is better suited for each query. This leads to a roughly 50% improvement in performance, reaching the individual upper bounds of IQR and IQE that we targeted in the beginning of this thesis. Through pruning techniques we have also reduced the number of options to roughly five from ten on average, maintaining the 50% improvement in performance. In other words, we have developed techniques to identify five options from the exponential number available that contain at least one option that is on average 50% better than the baseline. Development of techniques to further prune the number of options, or to combine them in some fashion that retains the high potential for improvement suggests the possibility of shifting the reformulation process for long queries from interactive to completely automatic. Given the techniques we have described, development of automatic techniques for long query reformulation is thus a promising avenue for future work. We further identified the queries in our training set that were not improved by all of IQR, IQE and SIRE, and analyzed them to motivate future directions for research.

CHAPTER 10
OTHER INTERACTION TECHNIQUES

In past chapters we introduced, evaluated, and optimized the IQR and IQE user interaction techniques. In this chapter we present other novel interaction techniques we explored. The techniques include topic selection, phrase identification, pattern extraction and negative feedback on named entities. The goal of these techniques is similar to that of IQR and IQE, i.e. collecting feedback from users in the form of responses to simple questions to improve a wider range of queries compared to automatic techniques. The experiments presented in this chapter are meant to be exploratory in nature and provide direction for future work. More rigorous experiments and user studies are required to completely validate the described techniques.

10.1 Topic Selection

Figure 10.1 provides a pictorial description of the effect (difference) on average precision for a set of 50 queries before and after using PRF. Clearly, the effect of PRF is lopsided. We believe the assumption that the top n documents from the initial retrieval is relevant and provide a good model of the targeted topic failed for some topics. To tackle this problem we considered expanding our search to an external corpus, i.e. a corpus different from the one we were currently searching. We hoped that the external corpus would have a greater number of documents relevant (and retrievable) to the query so that a better relevance model could be created. By doing so, we hoped to prevent the lopsided of effect of PRF, and obtain more uniform improvements. Previous work by Diaz and Metzler (2006) demonstrated the utility

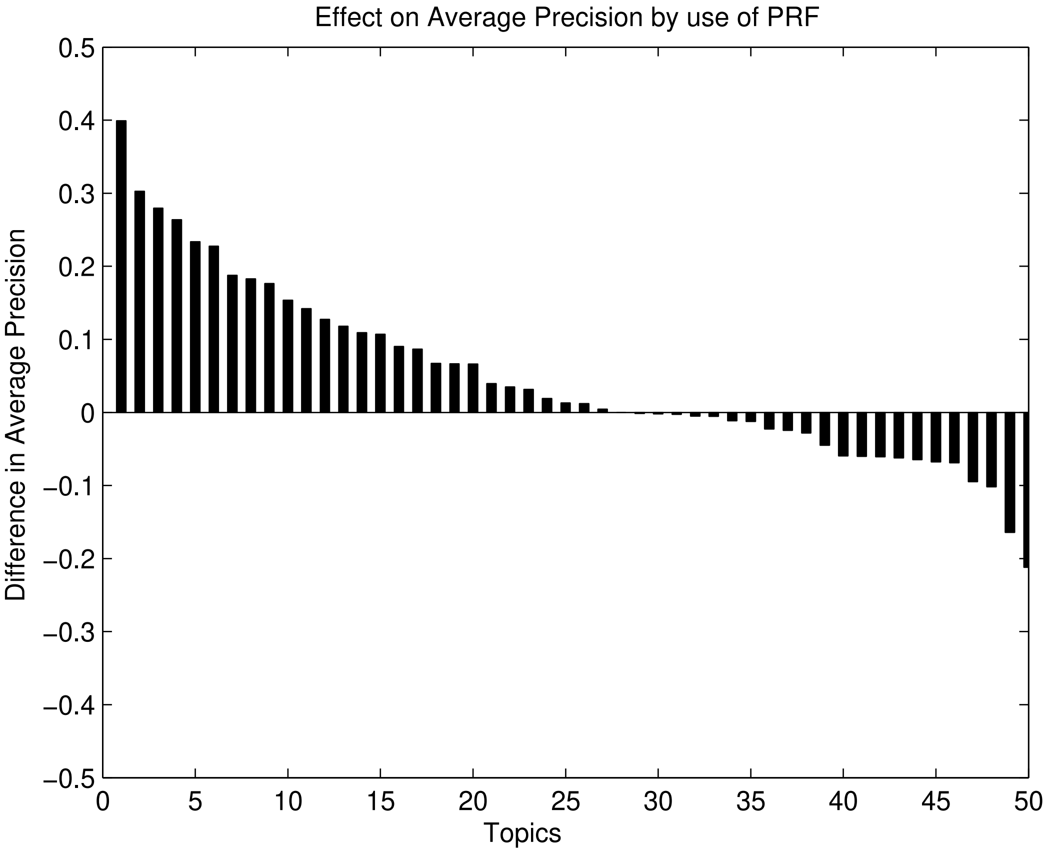


Figure 10.1. The difference in performance due to query expansion using PRF over the baseline query. Queries to the left experienced improvements in performance while those to the right ended up being worse than baseline.

of using a large external corpus to automatically improve the PRF language model, and showed that a 10% improvement in MAP stable across more topics was possible. However, the improvements in performance were dependent on the quality of the external corpus used for expansion. Our work is different as we involve the user in building a better language model through topic selection. The advantage of doing so is that the user can select or reject candidates for expansion leading to more stable improvements in performance across queries.

There has also been past work (Anick and Tipirneni, 1999; Cutting et al., 1992) in providing users a set of terms/phrases to describe the different clusters or aspects in a result set or query. The terms/phrases were used to present to users a quick summary of groups of documents, or provide options to refine queries. Our approach (Kumaran and Allan, 2006a) was to utilize a human-generated list of topics - the names of USENET NewsGroups, and ask the user to select group(s) they would expect to find their query discussed in. Since USENET newsgroups are human-maintained, and focused on particular topics, they had the potential to act as rich models of topics relevant to a user's query. We refer to this technique as *Topic Selection*.

10.1.1 Experimental Setup

We used the same set of topics and collections we used for IQR and IQE experiments: TREC Robust 2004 and Robust 2005. The only difference was that we used a randomly selected subset of 50 queries from the 250 queries in the Robust 2004 track. Our baseline queries consisted of terms from the title and description portions of the queries. As our retrieval system, we used version 2.2 of the open-source Indri system. We used the 418 stop words included in the stop list used by the InQuery system, and the K-stem stemming algorithm implementation provided as part of Indri.

Our baseline system (QL) is a query-likelihood variant of statistical language modeling. The pseudo-relevance feedback technique is based on relevance models (Lavrenko and Croft, 2001). We used the top 25 documents for feedback, and appended 25 terms to the original query.

In response to each query, we used the Indri search engine to search through a USENET archive spanning twenty years, and returned the *titles* of the newsgroups occurring in the top two hundred results. There was considerable overlap between the newsgroups the results belonged to. We ordered the newgroups by the number of results associated with each of them, and selected at most twenty to present to the user. Users were instructed to select the newsgroup(s) they believed would discuss their query using the titles as a guide. Once the user selected the titles, we restricted

		QL	PRF	Topic Selection
Robust05	MAP	0.2278	**0.2750**	**0.3027**
	GMAP	0.1533	0.1541	0.1777
	Queries Improved	–	54%	68%
Robust04	MAP	0.3418	**0.3622**	0.3695
	GMAP	0.2155	0.2079	0.2451
	Queries Improved	–	29%	33%

Table 10.1. Performance of the different systems in terms of MAP, GMAP, and percentage of queries improved. Entries in bold face are statistically significant improvements (paired t-test,$\alpha = 0.05$) . The Topic Selection system was compared with PRF for statistical significance, while PRF was compared with QL.

the query to the particular group(s), and used the results as a topic model for the query. Using this topic model, as well as the collection model, we performed PRF.

10.1.2 Results

An overview of the experimental results is provided in Table 10.1. We notice that PRF improves over QL in both cases, but the improvement across queries is not uniform. The results for the Robust04 queries are even more lopsided: a gain in MAP was achieved by improving less than a third of the queries, and even showing a drop in GMAP. We can observe that using USENET group information improves over PRF in both collections. The improvement in GMAP indicates that the Topic Selection system succeeded in bringing about an improvement across many more queries than PRF. Figure 10.2 provides a pictorial description of the changes in AP across queries due to Topic Selection. For comparison with PRF, Figure 10.1 is overlaid. The effect of using Topic Selection is less lopsided than using PRF.

10.1.3 Conclusions

Simple inputs from the user, in this case topic selection, can help improve performance beyond the state-of-the-art. The use of external sources of information helped build better topic models for selecting expansion terms. This way, improvement was

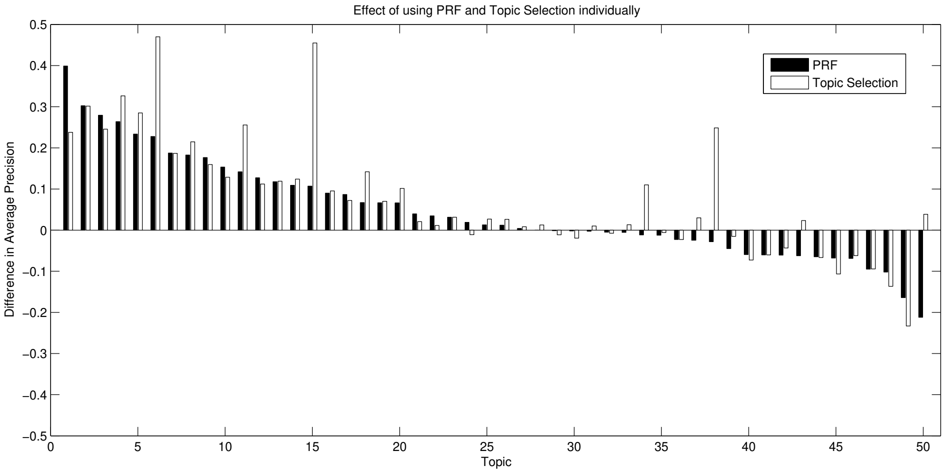


Figure 10.2. Comparison of the difference in performance due to query expansion using PRF and Topic Selection, over the baseline query.

obtained by not just doing better on easier queries, but by doing better on a larger set.

10.2 Phrase Identification

The use of phrases (Croft et al., 1991; Fagan, 1987) in queries is known to enhance precision in a number of Information Retrieval tasks. Consider the case of the query *What efforts have been made to stabilize the Leaning Tower of Pisa?*. When used as a simple query, there is possibility that a number of documents with references to *leaning*, the more common of the two terms, will be retrieved even though the term *Pisa* might not exist. If we instead search with the phrase "Leaning Tower of Pisa", we can ensure that only documents about the tower are retrieved, omitting spurious matches and improving precision. Consider the following examples.

> *Topic 320*: Fiber optic link around the globe (Flag) will be the *world's longest* undersea fiber optic cable. Who's involved and how extensive is the technology on this system. What problems exist?

> *Topic 339*: What drugs are being used in the treatment of *Alzheimer's Disease* and how successful are they?

In the two examples above the apostrophe is used to form possessives of nouns. In such situations we can expect the two terms to occur as a phrase in the relevant documents too.

> *Topic 344*: What steps have been taken *world-wide* by those bearing the cost of *E-mail* to prevent excesses?
>
> *Topic 443*: What is the extent of U.S. (government and private) investment in *sub-Saharan* Africa?

In these two examples hyphens are used in different contexts to indicate compound words. Again, we can expect to see these terms appear either as phrases or mentioned as one word in relevant documents.

> *Topic 443*: Find documents that discuss issues associated with *so-called orphan drugs*, that is, drugs that treat diseases affecting relatively few people.

The use of quotation marks also implies that the enclosed terms form a phrase.
In many IR systems punctuation is discarded while parsing. Interestingly, as in the examples above show, there is utility in pre-processing. We can make use of punctuation to identify useful phrases in the query automatically.
With a view to improving precision, and believing that users might have some previous knowledge about the topic they are interested in, we asked the user to identify the phrases in the query (Kumaran and Allan, 2006a)[1]. This system is referred to as Phrases.

[1] Only around 40% of the test queries had identifiable phrases in them.

		QL	Phrases
Robust05	MAP	0.2278	0.2307
	GMAP	0.1533	0.1552
	Queries Improved	–	14%
Robust04	MAP	0.3418	0.3508
	GMAP	0.2155	0.2192
	Queries Improved	–	11%

Table 10.2. Performance of the Phrases system in terms of MAP, GMAP, and percentage of queries improved.

For these experiments, we asked a user to edit his query to specify phrases in the Indri query language. We did not perform automatic analysis of the queries to generate options the user could choose from. Once automatic analysis is incorporated too, we hypothesize that an actual interaction with the user could involve questions such as those given below.

> *Query*: Find documents that discuss the impact Prime Minister Margaret Thatchers' resignation may have on U.S. and U.K. relations.
>
> *Simple question*: Is it correct that you see Margaret Thatchers' resignation as a useful phrase?

10.2.1 Experimental Setup

The experimental setup for this interaction technique was the same as that for Topic Selection.

10.2.2 Results

The improvement due to use of phrases (Table 10.2) was not statistically significant. This could in part be due to the fact that less than 40% of queries had identifiable phrases in them. The numerical improvements we observed were in spite of the fact that the improvements due to this technique are averaged across all the queries - even those that did not have phrases in them.

10.2.3 Conclusions

The use of phrases contributed to improvements in retrieval performance, and is a promising avenue for interaction with the user. We expect the interaction to be light-weight, as the questions involve terms familiar to the user. Providing the user with some context (in a fashion similar to IQR and IQE) will potentially help make the decision of choosing appropriate phrases easier.

10.3 Pattern Extraction

Certain term patterns occur frequently in particular topics. For example, in news reports on *bomb attacks*, a discerning reader can observe that the terms *killed* and *injured* occur frequently within a window of around eight terms.
We can view this technique, referred to as Patterns, as an extension to Phrases. Consider the query given below.

> *Query*: Identify hydroelectric projects proposed or under construction by country and location. Detailed description of nature, extent, purpose, problems, and consequences is desirable.

We could parse the top-ranked documents and request the user to provide an answer to a question like

> *Simple question*: Would you expect to see <u>three</u> and <u>dam</u> nearby, with terms such as <u>gorges</u> between them?

Our preliminary experiments involved processing the top-ranked documents from the USENET groups chosen by the user (Section 10.1) for each query and identifying the top ten most frequently appearing bigrams. These bigrams, along with the average distance between constituent terms interpreted as term window constraints, were appended to the original query. The resulting system was referred to as Patterns (Kumaran and Allan, 2006a).

		QL	Patterns
Robust05	MAP	0.2278	**0.2563**
	GMAP	0.1533	0.1633
	Queries Improved	–	56%
Robust04	MAP	0.3418	0.3508
	GMAP	0.2155	0.3375
	Queries Improved	–	21%

Table 10.3. Performance of the Patterns system in terms of MAP, GMAP, and percentage of queries improved.

10.3.1 Experimental Setup

The experimental setup for this interaction technique was the same as that for Topic Selection.

10.3.2 Results

The expansion of the query with bigram patterns resulted in statistically significant improvements over the baseline system on one of the collections (Table 10.3). Improvements in both MAP and GMAP were recorded.

10.3.3 Conclusions

We hypothesize that user interaction using this technique will be more difficult than previous techniques. This is because the user will have to provide feedback on terms they are possibly not familiar with. For example, to answer the question in the example given in this section, the user will have to know that the *Three Gorges Project* is a hydroelectric project in China. While providing some context might help to an extent, we do not have experimental results as yet proving this hypothesis.

10.4 Negative Feedback on Named Entities

Queries in template form are a new paradigm in the way users can convey complex information needs to search engines. Templated queries find most use when the same type of information is repeatedly queried about. The following templated query from the TREC 2006 complex Interactive Question Answering (ciQA) (Dang et al., 2006) track is one such example. It consists of two parts, the template itself and a narrative.

> Template: What evidence is there for transport of [drugs] from [Bonaire] to [the United States]?
>
> Narrative: The analyst would like to know of efforts made to discourage narco traffickers from using Bonaire as a transit point for drugs to the United States. Specifically, the analyst would like to know of any efforts by local authorities as well as the international community.

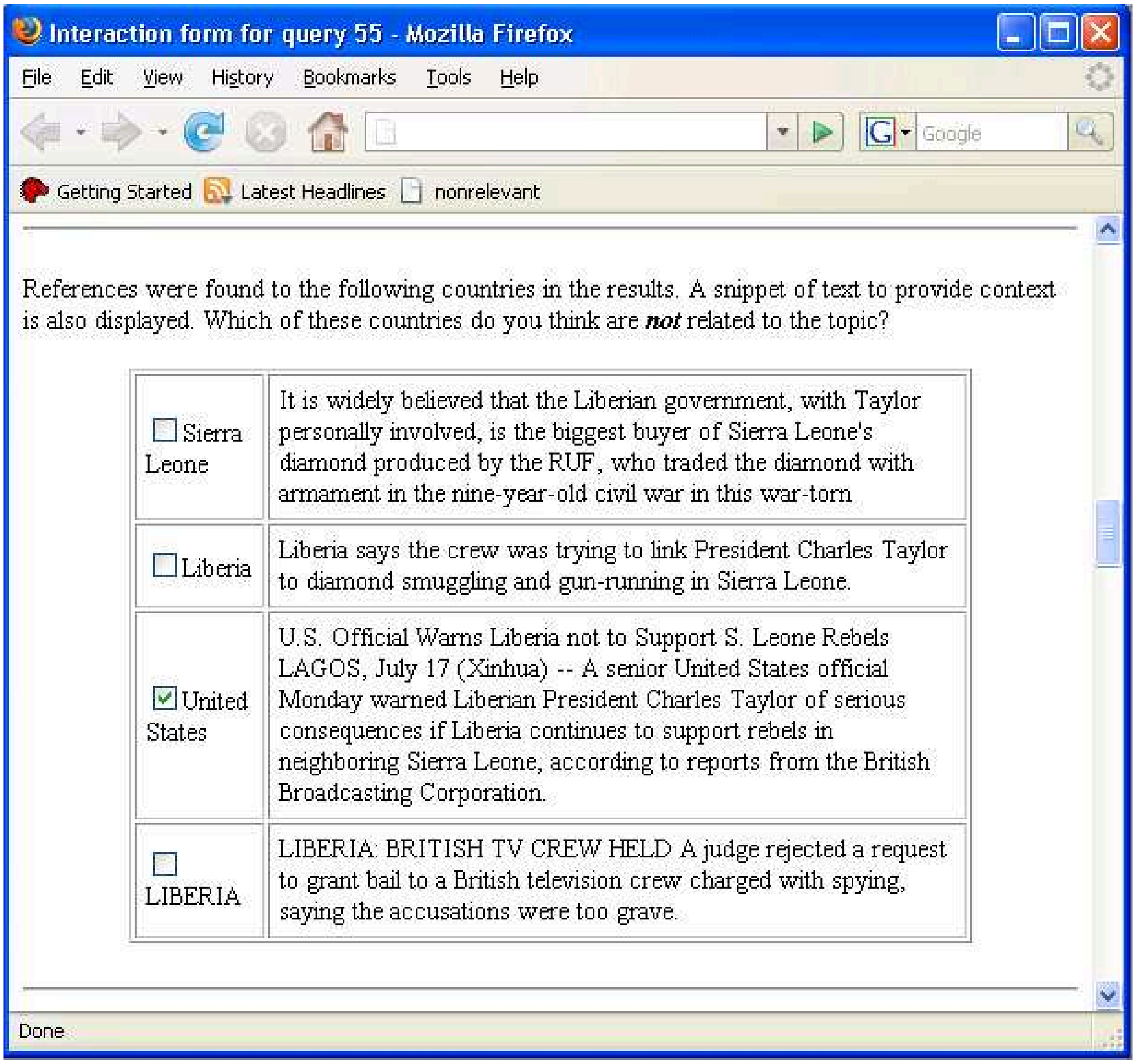
A user interested in the general topic of transfer of certain goods from one location to another can use this template by simply instantiating the free slots with the goods and locations she is interested in. To further expatiate upon the information need, the user can also include free-form text in the narrative section. Once the user has instantiated a template, the next task is to convert this templated query into one understandable by a search engine. Using the entire query in its original form can result in poor effectiveness due to the presence of extraneous terms like *analyst*, *specifically*, and *international community.* Hence there is need for an intermediary system, or a query processor, to convert this complex information need into an effective query in the search engine's query language. In creating this effective query, additional information about the template itself can be leveraged. For example, for the template given above, we can create a query that specifies that the locations Bonaire and United States should co-occur in a document, and that the term drugs should be expanded with a list of drugs. The possibility of incorporating template-specific features sets templated querying apart from the domain of free-form searching. Also, unlike ad hoc retrieval, the output is not a list of ranked documents, but a set of *snippets* of text extracted from documents, in decreasing order of relevance.

There has recently been great interest in utilizing annotations in data provided by the Information Extraction community for Information Retrieval. We took a step towards that direction by making use of named entities identified using BBN Identifinder (Bikel et al., 1999). To also move away from the usual procedure of identifying named entities and folding them into the query, we decided (Kumaran and Allan, 2006b) to instead use them for post-retrieval snippet processing. This was motivated by the observation in training data that often unrelated people, organizations and locations were present in the results. Since it was not possible to predict which named entities were unrelated, we decided to involve the user in making that decision. We believed that such information could be used to *clean up* the final results. To this end, we provided the user with separate lists of people, locations, and organizations identified in the final output from our baseline run. Along with each named entity, we displayed a sentence in which it occurred to provide some context to help the user make a decision whether the named entity was *not* relevant to the information need. We hoped that this kind of *negative* feedback (Cool et al., 1996) would favorably impact the precision of our results.

Figure 10.3 is a screen shot of a section of the interface requesting information from the user. The query template was *Is there evidence to support the involvement of [Charles Taylor] in [diamond smuggling]?*.

10.4.1 Experimental Setup

The test collection was the one-million document AQUAINT collection. We used version 2.3.2 of the open source Indri retrieval system. Since we had to return snippets as results instead of documents, we created a pipelined system consisting of an Indri document retrieval system, sentence segmenters and named-entity identifiers. Further details on the system are available in Kumaran and Allan (2006b).



Interaction form for query 55 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Google
Getting Started
Latest Headlines
nonrelevant
Done

References were found to the following countries in the results. A snippet of text to provide context is also displayed. Which of these countries do you think are ***not*** related to the topic?

☐ Sierra Leone	It is widely believed that the Liberian government, with Taylor personally involved, is the biggest buyer of Sierra Leone's diamond produced by the RUF, who traded the diamond with armament in the nine-year-old civil war in this war-torn
☐ Liberia	Liberia says the crew was trying to link President Charles Taylor to diamond smuggling and gun-running in Sierra Leone.
☑ United States	U.S. Official Warns Liberia not to Support S. Leone Rebels LAGOS, July 17 (Xinhua) -- A senior United States official Monday warned Liberian President Charles Taylor of serious consequences if Liberia continues to support rebels in neighboring Sierra Leone, according to reports from the British Broadcasting Corporation.
☐ LIBERIA	LIBERIA: BRITISH TV CREW HELD A judge rejected a request to grant bail to a British television crew charged with spying, saying the accusations were too grave.

Figure 10.3. A section of the interface requesting user feedback on *location* entities

	MAP
Baseline	0.061
Interactive	0.081 (0.067)

Table 10.4. Comparison of the MAP scores for each run. The value in brackets is the p-value from a two-tailed paired t-test comparing the run with the baseline

10.4.2 Results

Table 10.4 compares the MAP scores of the baseline and interactive runs. The baseline results are quite poor, exposing the limitations of adapting an IR system to perform question answering. However, of interest is the fact that precision of the interactive run is higher. Though we deployed multiple interaction techniques, after analyzing the results from different interactive runs, we believe that this is the effect of using the negative feedback provided by the users in the form of non-relevant named-entity identification. The problems caused by the observed failure of other interaction techniques appears to have dampened the positive effect of named-entity feedback.

10.4.3 Conclusions

It would have been more informative to compare measures like p@5 and p@10. Unfortunately, since the evaluation was performed by an external agency, National Institute of Standards and Technology, we were unable to perform further analysis. However, the numbers still show the utility of feedback on named entities.

10.5 Summary

In summary, there is sufficient evidence to suggest that the sort of simple user interaction user strategies we have described in this chapter are worth pursuing, and are an important research goal to facilitate better performance in information retrieval tasks. The Topic Selection and Phrase Extraction techniques resulted in significant improvements over the baseline in terms of MAP and GMAP, indicating that these techniques achieve the original goal of improving performance for a wider range of queries. Further exploration is required to determine if the same set of queries are improved by both techniques. If that is not the case, then techniques to similar to SIRE need to be developed to selectively invoke Topic Selection and Phrase Extraction. This way greater improvements in performance can possibly be achieved.

CHAPTER 11

CONCLUSIONS AND FUTURE WORK

11.1 Conclusions

We began this thesis by demonstrating the vast potential for improved information retrieval through user interaction for long queries. The long queries (Chapter 4) could be in the form of either a large amount of text entered by a user or the automatically expanded version of a short query. In successive chapters we designed an interface for effective user interaction and demonstrated its efficacy (Chapter 6), and developed procedures for selective (Chapter 7), efficient (Chapter 8), and reduced (Chapter 9) user interaction. Our contribution to the field of information retrieval through the development and analysis of these and related ideas include:

1. **Bringing to focus long queries**. With most queries issued to web search engines being two to four terms in length, most research focus has been on handling short queries. We have demonstrated the potential of long queries, and developed techniques to realize them.

2. **A new look-ahead user interface**. We have developed a user interface well-suited for rapid analysis and selection of query reformulations. Feasibility studies revealed the utility of the interface in not only rapidly directing users towards their information needs, but also providing them a preview of the collection contents and a sense of how the retrieval system works.

3. **Selective user interaction**. The same problems that plague automatic techniques are prevalent in interactive techniques too: i.e., user interaction has

the potential to lead to improvements only for a subset of queries. We have presented techniques to identify situations where user interaction has no utility. These techniques will potentially have a bearing on the adoption of user interaction in mainstream information retrieval systems.

4. **Efficient user interaction**. The often-ignored aspect of interactive retrieval relates to the processes that are used to analyze and generate options to present to users. To the best of our knowledge, the technique of observing the properties of ideal options and using them to inform the search for options is unique to our work, and can be easily extended to the generation processes for interactive techniques.

5. **Optimal user interaction**. Another important contribution to the field is tackling the problem of determining the optimal number of options to present to users. The easily-generalizable techniques we have developed not only reduce the number of options shown to users but also retain the important, showing that users can obtain the same improvements in performance by interacting *less* with an IR system.

6. **Combining interactive techniques**. The need to invoke the right interactive technique for the right information need is well known. We have shown that we can get the user to implicitly guide selection of the right interactive technique, and demonstrated the high performance gains of doing so.

11.2 Future Work

The motivation for continued work is best summed up by Table 11.1. It shows the extent to which the techniques we have described in this thesis have taken us (from a MAP of 0.240 to 0.302), and the remaining improvement (from a MAP of 0.302 to 0.366) that still is open for further research to achieve.

System	P@5	P@10	NDCG@15	MAP
Baseline (QL)	0.472	0.397	0.379	0.240
Techniques in this thesis	**0.634**	**0.527**	**0.499**	**0.302**
Upper Bound	**0.799**	**0.671**	**0.626**	**0.366**

Table 11.1. Comparison of performance for 249 TREC 2004 Robust track *description* queries. Values in bold are significantly better than PRF. Statistical significance was measured using a paired t-test, with α set to 0.05

11.2.1 Improved IQE and IQE

11.2.1.1 Techniques to generate options

A number of avenues exist within the framework of the IQE and IQR techniques that can potentially lead to improved performance. With regard to improving the quality of options presented to user, parsing long queries using NLP techniques is a promising direction.

Figure 11.1 provides the potential for improvement in AQE using two variants of automatic selection techniques described in Section 5.1. While the first one, *Total*, selects sub-queries to present to the user based on the weight of the maximum spanning tree, the second, *Average*, uses the same score but divided by the number of terms in the sub-query. We notice that using *Total* results in at least one of the top ten sub-queries shown to the user being better than or equivalent to the baseline. Using *Average* has the benefit of higher potential for improvement on a fraction of queries, without ensuring at least one sub-query choice equivalent to or better than the baseline for all queries[1]. Developing hybridized versions of selection techniques is thus useful.

11.2.1.2 Option pruning techniques

While the option analysis procedure described in Section 8.3 doesn't involve any querying of the index, the option-pruning procedure (Section 8.2) requires querying

[1] The curves have been generated by ordering the potential improvements in descending order. Comparison on a per-query basis is not possible with Figure 11.1

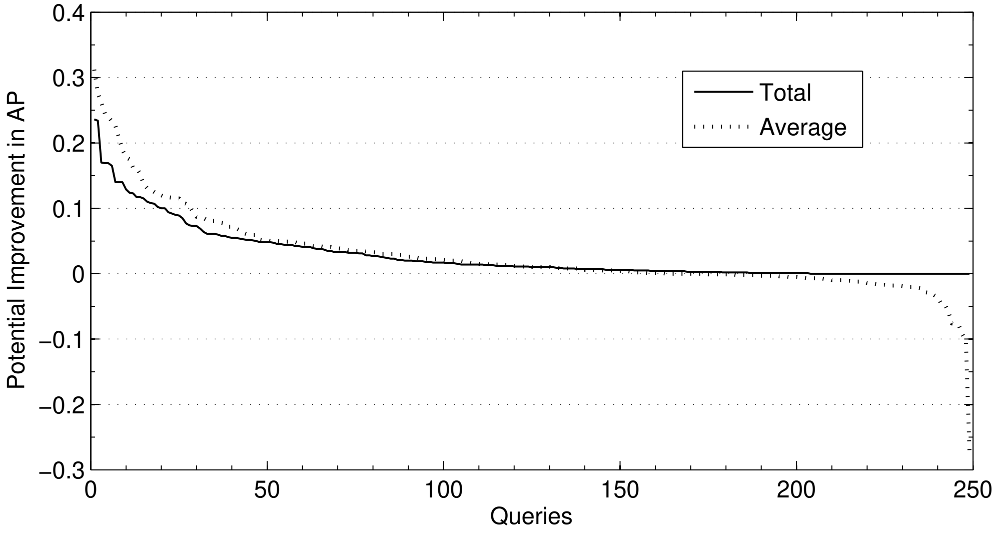


Figure 11.1. The utility of two automatic options selection techniques

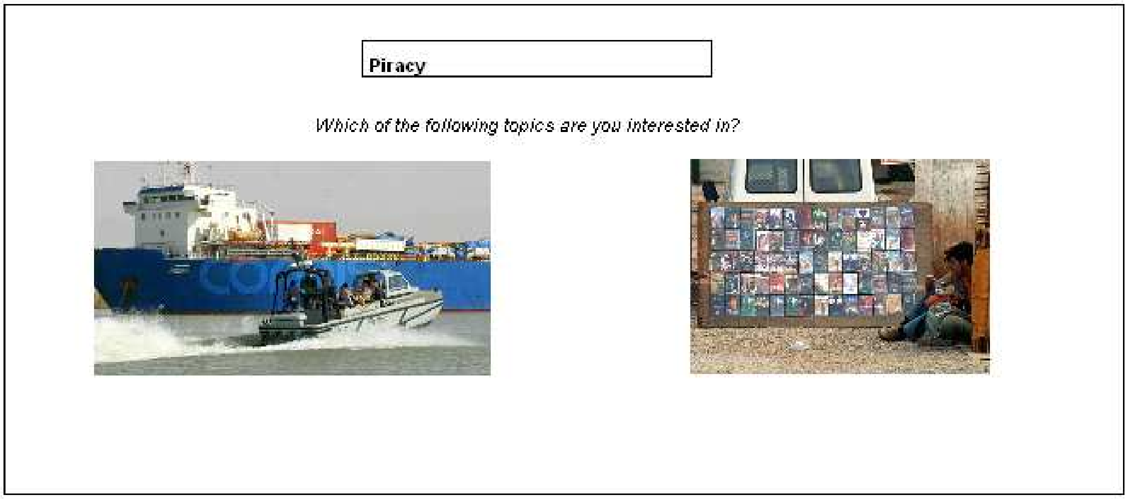
the index to obtain top-ranked documents. A better technique to approximate the top-ranking documents will make the process more efficient.

11.2.1.3 IQR + AQE

IQR results in a concise version of the long query that is potentially better. In future work we plan to explore using the query identified thorough IQR as a starting point for either automatic or interactive query expansion. This two-stage interaction could potentially improve effectiveness even further.

11.2.2 More interaction techniques

IQE and IQR are not exclusive techniques to handle long queries. Interaction with the user can occur at various stages of the retrieval process. In the query formulation stage, the user's query can be analyzed to suggest modifications. Knowledge of the collection content can also be leveraged to provide the user with evidence to either expand or relax a query. We propose the following interaction techniques to further handle long queries.



Piracy

Which of the following topics are you interested in?

Figure 11.2. A hypothetical interface for query disambiguation using images.

- Query disambiguation using images For example, for a query *piracy*, a potential way to interact would be as shown in Figure 11.2.

- Query cohesiveness Quite often queries contain terms that hamper effective search. Removing these terms or down weighting them results in considerable improvement in performance. Previous research (Kumaran and Allan, 2007) can be extended to perform a better job of guiding the user in selecting the right terms for use from their query.

- Query quality Providing the user with pictorial feedback in the form of an on line pie chart showing the percentage of the corpus affected by addition or removal of query terms could potentially guide the user in determining the best set of terms to use as a query.

- Targeting named entities To find the best *Best Retirement Country*, if a user specifies from a list of named entities that she is interested in *location* named entities, off-topic results like those on retirement funding can be potentially avoided.

Opportunities for interaction also exist during the retrieval process. In response to a query the system can interact with the user to guide the search. To this end, we plan to explore involving the user to provide guidance in the following aspects.

- Entity context Terms and entities can carry different meanings in different contexts. A free-form query ignores this fact, depending on other terms in a query to provide context. Since term-weighting can be inaccurate and is the cause for many retrieval failures, it is useful to have a mechanism to further clarify the context a term or entity is used in. For example, the term *Bonaire* could appear in a document as

    - part of an address, as in Bonaire, Netherlands Antilles
    - a location, as in Bonaire
    - or an organization, as in Bonaire Democratic Party

  Feedback from the user about the context can potentially focus the query better.

- Person named entities Knowing that Michael Gorbachev was the President of the USSR during the Soviet Union's withdrawal from Afghanistan boosts the quality of the results for the TREC query *Soviet withdrawal Afghanistan*. This opens an opportunity for a quick round of interaction that involves displaying the person named entities found in the top results from an initial run and asking the user to choose the persons she believes could be related to the query. A very short biography from a source like Wikipedia could help the user make the decision.

- Selecting patterns found in the top-ranked documents. The patterns can be found using motif-finding algorithms adapted from bioinformatics research.

- Top-ranked sentences The user can provide feedback on the results returned by the system. While earlier work (Cool et al., 1996) in negative feedback involved simply assigning negative weights to candidate terms, our research (Kumaran and Allan, 2006b) suggests that negative feedback involving named entities might be useful in improving the precision of the results. Displaying top-ranked sentences has been found to be an effective way of summarizing results. We could take this further and ask the user to select non-relevant sentences. The negative feedback could then be user to clear the results list or reformulate the query.

BIBLIOGRAPHY

Aalbersberg, I. J. (1992). Incremental relevance feedback. In *SIGIR '92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 11–22, New York, NY, USA. ACM.

Al-Maskari, A., Sanderson, M., and Clough, P. (2007). The relationship between ir effectiveness measures and user satisfaction. In *SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 773–774, New York, NY, USA. ACM.

Allan, J. (2003). The HARD Track Overview in TREC 2003. High Accuracy Retrieval from Documents. In *TREC 12 Proceedings*.

Allan, J., Callan, J. P., Croft, W. B., Ballesteros, L., Broglio, J., Xu, J., and Shu, H. (1996). INQUERY at TREC-5. In *Proceedings of The Fifth Text REtrieval Conference*, pages 119–132.

Allan, J., Leuski, A., Swan, R., and Byrd, D. (2001). Evaluating combinations of ranked lists and visualizations of inter-document similarity. *Information Processing and Management*, 37(3):435–458.

Anick, P. G. and Tipirneni, S. (1999). The paraphrase search assistant: terminological feedback for iterative information seeking. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 153–159, New York, NY, USA. ACM.

Belkin, N. J. (1993). Interaction with texts: Information retrieval as information-seeking behavior. *Information Retrieval*, pages 55–66.

Belkin, N. J., Carballo, J. P., Cool, C., jeng Lin, S., Park, S., Rieh, S. Y., Savage-Knepshield, P. A., Sikora, C., Xie, H., and Allan, J. (1997). Rutgers' TREC-6 interactive track experience. In *Proceedings of the Sixth Text REtrieval Conference (TREC 1997)*, pages 597–610.

Belkin, N. J., Cool, C., Stein, A., and Thiel, U. (1995). Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(3):379–395.

Belkin, N. J., Oddy, R. N., and Brooks, H. M. (1982a). Ask for information retrieval: part i.: Background and theory. *International Journal of Human-Computer Studies*, 38(2):61–71.

Belkin, N. J., Oddy, R. N., and Brooks, H. M. (1982b). Ask for information retrieval: part ii.: Results of a design study. *International Journal of Human-Computer Studies*, 38(3):145–164.

Bikel, D. M., Schwartz, R., and Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231.

Borgman, C. L., Belkin, N. J., Croft, W. B., Lesk, M. E., and Landauer, T. K. (1988). Retrieval systems for the information seeker: can the role of the intermediary be automated? In *CHI '88: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 51–53.

Borlund, P. (2000). Experimental components for the evaluation of interactive information retrieval systems. *Journal of Documentation*, (1):71–90.

Borlund, P. and Ingwersen, P. (1997). The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation*, (53(3)):225–250.

Bruza, P., McArthur, R., and Dennis, S. (1998). Searching the world wide web made easy? The cognitive load imposed by query refinement mechanisms. In *Third Australian Document Computing Symposium*, pages 65–71.

Buckley, C. (2004). Why current ir engines fail. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 584–585, New York, NY, USA. ACM Press.

Buckley, C., Mitra, M., Walz, J., and Cardie, C. (2000). Using clustering and superconcepts within smart: TREC 6. *Information Processing and Management*, 36(1):109–131.

Carmel, D., Yom-Tov, E., Darlow, A., and Pelleg, D. (2006). What makes a query difficult? In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 390–397, New York, NY, USA. ACM.

Chen, F., Gargi, U., Niles, L., and Schütze, H. (1998). Multi-modal browsing of images in web documents. In *Proceedings of the SPIE Conference on Document Recognition and Retrieval*, pages 122–133.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. volume 16, pages 22–29, Cambridge, MA, USA. MIT Press.

Cool, C., Belkin, N. J., and 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*, page 341, New York, NY, USA. ACM Press.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms, Second Edition*. The MIT Electrical Engineering and Computer Science Series. The MIT Press.

Croft, W. B. and Thompson, R. H. (1987). I3R: a new approach to the design of document retrieval systems. *Journal of the American Society for Information Science*, 38(6):389–404.

Croft, W. B., Turtle, H. R., and Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. In *SIGIR '91: Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 32–45, New York, NY, USA. ACM.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *SIGIR '02: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, New York, NY, USA. ACM Press.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2004). A framework for selective query expansion. In *CIKM '04: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pages 236–237, New York, NY, USA. ACM.

Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR '92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, New York, NY, USA. ACM.

Dang, H. T., Lin, J., and Kelly, D. (2006). Overview of the TREC 2006 question answering track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*.

Diaz, F. and Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161, New York, NY, USA. ACM.

Doyle, L. (1959). Programmed interpretation of text as a basis for information retrieval systems. In *Proceedings of the Western Joint Computer Conference*, pages 60–63, San Francisco, USA.

Doyle, L. (1975). *Information Retrieval and Processing*.

Dumais, S., Cutrell, E., and Chen, H. (2001). Optimizing search by showing results in context. In *CHI '01: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 277–284.

Fagan, J. (1987). Automatic phrase indexing for document retrieval. In *SIGIR '87: Proceedings of the 10th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 91–101, New York, NY, USA. ACM.

Harman, D. (1988). Towards interactive query expansion. In *SIGIR '88: Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–331, New York, NY, USA. ACM.

Harman, D. (1991). How effective is suffixing? *Journal of the American Society of Information Science*, 42(1):7–15.

Harman, D. and Buckley, C. (2004a). The NRRC reliable information access (RIA) workshop. In *SIGIR '04: Proceedings of the 27th ACM SIGIR Conference*, pages 528–529.

Harman, D. and Buckley, C. (2004b). The NRRC reliable information access (RIA) workshop. In *SIGIR '04: 27th ACM SIGIR*, pages 528–529.

He, B. and Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *The Eleventh Symposium on String Processing and Information Retrieval (SPIRE)*.

Hearst, M. A. (1995). Tilebars: Visualization of term distribution information in full text information access. In Irvin R. Katz, Robert L. Mack, L. M. M. B. R. J. N., editor, *Proceedings of the ACM CHI 95 Human Factors in Computing Systems Conference*, pages 59–66.

Hearst, M. A. and Karadi, C. (1997). Cat-a-cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *SIGIR '97: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–255, New York, NY, USA. ACM Press.

Hendley, R. J., Drew, N. S., Wood, A., and Beale, R. (1995). Narcissus: Visualizing information. In *Proceedings of the 1995 Information Visualization Symposium*, pages 90–96.

Henninger, S. and Belkin, N. J. (1996). Interface issues and interaction strategies for information retrieval systems. In *CHI '96: Conference Companion on Human Factors in Computing Systems*, pages 352–353.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Kaki, M. (2004). Proportional search interface usability measures. In *NordiCHI '04: Third Nordic Conference on Human-Computer Interaction*, pages 365–372.

Kaki, M. (2005). Findex: search result categories help users when document ranking fails. In *CHI '05: Proceedings of the SIGCHI Conference on Human factors in computing systems*, pages 131–140, New York, NY, USA. ACM Press.

Kelly, D. and Belkin, N. J. (2004). Display time as implicit feedback: understanding task effects. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 377–384, New York, NY, USA. ACM.

Kelly, D., Dollu, V. D., and Fu, X. (2005). The loquacious user: a document-independent source of terms for query expansion. In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 457–464, New York, NY, USA. ACM.

Kelly, D. and Teevan, J. (2003). Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28.

Kern, J. P., Pattichis, M., and Stearns, S. D. (2003). Registration of image cubes using multivariate mutual information. In *Thirty-Seventh Asilomar Conference*, volume 2, pages 1645–1649.

Koenemann, J. and Belkin, N. J. (1996). A case for interaction: a study of interactive information retrieval behavior and effectiveness. In *CHI '96: Proceedings of the SIGCHI Conference on Human factors in computing systems*, pages 205–212, New York, NY, USA. ACM Press.

Kraft, R., Chang, C. C., Maghoul, F., and Kumar, R. (2006). Searching with context. In *WWW '06: 15th International Conference Proceedings*, pages 477–486.

Krovetz, R. (1993). Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 191–202, New York, NY, USA. ACM.

Kumar, H. P., Plaisant, C., and Shneiderman, B. (1997). Browsing hierarchical data with multi-level dynamic queries and pruning. *International Journal of Human-Computer Studies*, 46(1):103–124.

Kumaran, G. and Allan, J. (2006a). Simple questions to improve pseudo-relevance feedback results. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 661–662, New York, NY, USA. ACM.

Kumaran, G. and Allan, J. (2006b). Umass at TREC ciQA. In *15th Text REtrieval Conference (TREC 2006).*

Kumaran, G. and Allan, J. (2007). A case for shorter queries, and helping users create them. In *Human Language Technologies 2007: The Conference of the North*

American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pages 220–227, Rochester, New York. Association for Computational Linguistics.

Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, New York, NY, USA. ACM.

Leouski, A. and Allan, J. (1998). Visual interactions with a multidimensional ranked list. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 353–354, New York, NY, USA. ACM.

Leuski, A. and Allan, J. (2000). Lighthouse: Showing the way to relevant information. In *Proceedings of the IEEE Symposium on Information Visualization 2000*, pages 125–130.

Maron, M. and Kuhns, J. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the Association of Computing Machinery*, 7:216–244.

Metzler, D. and Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing & Management*, 40(5):735–750.

Nordlie, R. (1999). "user revealment" - a comparison of initial queries and ensuing question development in online searching and in human reference interactions. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 11–18, New York, NY, USA. ACM.

Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2 edition.

Robertson, S. (2006). On gmap: and other transformations. In *CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 78–83, New York, NY, USA. ACM Press.

Rocchio, J. J. (1966). *Document Retrieval Systems: Optimization and Evaluation.* PhD thesis, Harvard University, Cambridge, MA, USA.

Salton, G. and Buckley, C. (1997). Improving retrieval performance by relevance feedback. pages 355–364.

Shen, X. and Zhai, C. (2005). Active feedback in ad hoc information retrieval. In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59–66, New York, NY, USA. ACM.

Smucker, M. D. and Allan, J. (2006). Find-similar: similarity browsing as a search tool. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 461–468, New York, NY, USA. ACM.

Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *CIKM '99: Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 316–321, New York, NY, USA. ACM.

Spink, A., Goodrum, A., Robins, D., and Wu, M. M. (1996). Elicitations during information retrieval: implications for ir system design. In *Proceedings of the 19th ACM SIGIR Conference*, pages 120–127.

Spink, A., Jansen, B. J., Wolfram, D., and Saracevic, T. (2002). From e-sex to e-commerce: Web search changes. *Computer*, 35(3):107–109.

Stiles, H. (1961). The association factor in information retrieval. *Journal of the Association of Computing Machinery*, 8:271–279.

Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: A language model-based search engine for complex queries. In *International Conference on Intelligence Analysis*.

Swan, R. and Allan, J. (2000). Timemine (demonstration session): visualizing automatically constructed timelines. In *SIGIR '00: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 393, New York, NY, USA. ACM Press.

Swan, R. C. and Allan, J. (1998). Aspect windows, 3-d visualizations, and indirect comparisons of information retrieval systems. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 173–181, New York, NY, USA. ACM Press.

Tombros, A., Jose, J. M., and Ruthven, I. (2003a). Clustering top-ranking sentences for information access. In Koch, T. and Solvberg, I. T., editors, *Proceedings of the 7th European Conference on Digital Libraries. ECDL 2003.*, pages 523–528. Lecture Notes in Computer Science, Springer.

Tombros, A., Jose, J. M., Ruthven, I., and White, R. W. (2003b). Clustering the information space using top-ranking sentences: A study of user interaction. In Rauterberg, G. M., Menozzi, M., and Wesson, J., editors, *Proceedings of the 9th IFIP TC13 International Conference on Human-Computer Interaction. INTERACT 2003*, pages 928–931.

Turtle, H. and Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, 9(3):187–222.

Vassilvitskii, S. and Brill, E. (2006). Using web-graph distance for relevance feedback in web search. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 147–153, New York, NY, USA. ACM.

Vinay, V., Cox, I. J., Milic-Frayling, N., and Wood, K. (2006). On ranking the effectiveness of searches. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 398–404, New York, NY, USA. ACM.

Voorhees, E. M. (2006). The TREC 2005 robust track. *SIGIR Forum*, 40(1):41–48.

Voorhees, E. M. and Harman, D. (1996). Overview of the fifth text retrieval conference (TREC 5). In *TREC 5 Proceedings*.

White, R. W., Jose, J. M., and Ruthven, I. (2005). Using top-ranking sentences to facilitate effective information access. *Journal of the American Society for Information Science and Technology*, 56(10):1113–1125.

White, R. W. and Ruthven, I. (2006). A study of interface support mechanisms for interactive information retrieval. *Journal of the American Society for Information Science and Technology*, 57(7):933–948.

Xu, J. and Croft, W. B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112.

Zhou, Y. and Croft, W. B. (2006). Ranking robustness: a novel framework to predict query performance. In *CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 567–574, New York, NY, USA. ACM.