Adapting Information Retrieval Systems to User Queries

Giridhar Kumaran and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003, USA
{giridhar,allan}@cs.umass.edu

Abstract

Users enter queries that are short as well as long. The aim of this work is to evaluate techniques that can enable information retrieval (IR) systems to automatically adapt to perform better on such queries. By adaptation we refer to (1) modifications to the queries via user interaction, and (2) detecting that the original query is not a good candidate for modification. We will show that the former has the potential to improve mean average precision (MAP) of long and short queries by 40% and 30% respectively, and that simple user interaction can help towards this goal. We observed that after inspecting the options presented to them, users frequently did not select any. We present techniques in this paper to determine beforehand the utility of user interaction to avoid this waste of time and effort. We show that our techniques can provide IR systems with the ability to detect and avoid interaction for unpromising queries without a significant drop in overall performance.

1 Introduction

The quality of queries submitted to Information Retrieval (IR) systems directly affects the quality of search results generated (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 IR system works. This lack of knowledge is usually coupled with another unknown: the contents of the collection being searched. A disconnect thus exists between what users enter as queries and the ideal representation required to retrieve the documents they want (Nordlie, 1999).

In this paper we are interested in two types of queries, those that are long and those that are short. Shorter queries are more pervasive than longer ones - especially in the web domain. The average query length is around 2.3 words (Spink et al., 2002), and poses the challenge of understanding the user's information need from such a limited expression of it (Example: *"Hurricane Katrina impact"*). Longer queries are encouraged by search engines such as Y!Q beta[1] and PowerSet[2]. The motivation for encouraging this type of querying (Example: *"What was the economic impact of Hurricane Katrina on Mississippi?"*) is that longer queries provide more information in the form of context (Kraft et al., 2006), and this additional information can be leveraged to provide a better search experience. However, handling such queries is a

[1] http://yq.search.yahoo.com/
[2] http://www.powerset.com/

challenge too. Longer queries contain a number of extraneous terms - terms that a user believes are vital to conveying her information need, but in reality hurt retrieval performance due to the way they are handled by the underlying retrieval model.

We are interested in exploring automatic and interactive ways by which we can adapt IR systems to effectively and efficiently handle queries, short or long, submitted by users. While a number of techniques for adaptation exist (Kumaran and Allan, 2006) we are particularly interested in those that target modifications to the queries input by the user. Modifications to queries, either in the form of expansion or relaxation, have been widely studied and known to contribute to significant improvements in performance. Automatic Query Expansion (AQE) refers to the process of including related terms to the original query, while automatic query relaxation (AQR) refers to the dropping or down-weighting of terms from the original query. While the former is well-suited for short queries, the latter is known to work well for longer queries (Xu and Croft, 1996; Kumaran and Allan, 2007a). In Section 2 we will show that when an IR system substitutes the original query with an ideal set of query terms obtained by relaxation or expansion, improvements in performance up to 40% and 30% respectively are achievable. Targeting this potential improvement we explored automatic techniques (Section 3) for identifying the ideal set of terms for relaxation and expansion. As the results of our experiments in Section 5 show, automatic techniques fail to identify the ideal set of terms for adaptation. To overcome that we explore their utility when used in conjuction with guidance from the user, i.e. adaptation with some simple user interaction. We refer to the interactive versions of expansion and relaxation as interactive query expansion (IQE) and interactive query relaxation (IQR) respectively. The results of user studies (Section 6) validated the utility of these techniques in adapting to the user's query.

Analysis of those results reveals that overall (average) improvements in performance are attributable to high gains on a small fraction of queries (Section 7). In other words, user interaction has the potential to lead to improvements only for a subset of queries. Frequently, none of the options selected by the automatic procedures and presented to the user were any better than the original query.

We believe that systems should be able to anticipate and adapt to 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).

In Section 8 we present an expanded version of our previous work (Kumaran and Allan, 2007b) on determining when to interact with the user to obtain explicit feedback. We consider two settings - IQR and IQE. We base the decision to interact on the potential for improved performance with user involvement. Our approach is to examine the properties of the set of options presented to the user. We hypothesize that useful sets of options will have distinguishing features from non-useful ones. We also believe that although such a method will not be perfect, users will be willing to trade a slight, but not significant drop in performance in return for a better search experience.

In this study we

- motivate the need for and utility of adapting to users' queries,
- develop automatic techniques for such adaptation,
- demonstrate the utility of involving the user in adaptation, and
- develop techniques to enable systems to identify and adapt to situations where user interaction is futile.

2 Motivation

In this section we provide illustrative examples[3], one each for relaxation and expansion, to show the utility of

- *Query Relaxation*: Choosing a good sub-set (sub-query) from a long query's terms.
- *Query Expansion*: Choosing a good sub-set (expansion subset) of the original set of terms suggested by an automatic query expansion procedure for a short query.

The queries used in the Text REtrieval Conference (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. Almost all 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. In this paper, we used the description as a surrogate for a long query, and the title as one for a short query.

2.1 Query Relaxation

Consider the following long (description) query for TREC Topic 324:

Define Argentine and British international relations.

When this query was issued to a search engine, the average precision (AP, Section 4) 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 1. It can be observed that there are seven different ways of re-writing the original query to attain better performance. The best query, also among the shortest, did not have a natural-language flavor to it. It however had an effectiveness almost 50% more than the original query, showing the immense potential for query relaxation.

2.2 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. However, if instead of just using all twenty-five terms, 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 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.

2.3 Analysis

We analyzed the relationship of terms in the sub-queries and expansion subsets with the original query. We made the following observations that informed techniques to identify good sub-queries and expansion subsets.

[3] Section 5.1 provides experimental validation

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 1: The results of using all possible subsets (excluding singletons) of the original query as queries, sorted by AP. The italicized query in the table is the original one. The query terms were stemmed and stopped.

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 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.

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 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, 2004), and the focus of efforts to isolate the key *concept* terms in queries (Buckley et al., 2000; Allan et al., 1996).

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 explored a variety of procedures to automatically adapt the system to the user's query. We expected that a good query would have the following properties.

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

B. *Coherency*: The terms that constituted 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.

3 Automatic Selection Techniques

Our goal is to identify a subset of the original query (for query relaxation) or expanded set (for expansion). We hypothesize that such a subset needs to be *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. We not list the measures based on term co-occurrence we investigated.

3.1 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:

$$\begin{aligned} I(X;Y) &= \sum_x \sum_y p(x,y) log \frac{p(x,y)}{p(x)p(y)} \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \quad (1)$$

$H(X)$ 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. The mutual information between two terms (Church and Hanks, 1989) can be calculated using Equation 2.

$$I(x,y) = log \frac{\frac{n(x,y)}{N}}{\frac{n(x)}{N}\frac{n(y)}{N}} \tag{2}$$

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 1 to:

$$I(X_1;X_2;X_3;...;X_N) = \sum_{i=1}^{N} (-1)^{i-1} \sum_{X \subset (X_1,X_2,X_3,...,X_N), |X|=k} H(X) \tag{3}$$

The calculation of multivariate information using Equation 3 was very cumbersome, and we instead worked with the approximation (Kern et al., 2003) given below.

$$I(X_1;X_2;X_3;...;X_N) = \sum_{i,j=\{1,2,3,...,N; i \neq j\}} I(X_i;X_j) \tag{4}$$

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 set 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**.

3.2 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 2 could have a negative value. We attempted to tackle this problem by creating a maximum spanning tree (MaxST) over the fully connected graph G, and using the weight of the identified tree as a measure of the candidate query's quality (van Rijsbergen, 1979). We used Kruskal's minimum spanning tree (Cormen et al., 2001) algorithm 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**.

3.3 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 relaxation, 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 relaxation that retained named entities as **NE_Average** and **NE_MasT**. For query expansion, we did not pursue expansion by only named entities.

4 Experimental Setup

We used version 2.5 of the Indri search engine, developed as part of the Lemur[4] project. 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. The pseudo-relevance feedback mechanism we used is based on relevance models (Lavrenko and Croft, 2001).

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

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 for our experiments. 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 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. User interaction held promise for improvement in these collections. The TREC 5 ad-hoc collections 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 do not stand 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.

We stemmed the collections using the Krovetz stemmer provided as part of Indri, and used a manually-created stoplist of twenty terms (*a, an, and, are, at, as, be, for, in, is, it, of, on, or, that, the, to, was, with* and *what*). 249 queries from the TREC Robust 2004 track were analyzed to determine and fine tune the procedure we developed to determine the utility of interaction. The remaining 150 queries, 50 each from the three remaining tracks, were used to test the effectiveness of various techniques we present in this paper.

For all systems, we report mean average precision (MAP) and geometric mean average precision (GMAP) (Robertson, 2006). MAP is the most widely used measure in Information Retrieval. 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 APs of a set of queries. Similarly, GMAP is the geometric mean of the APs 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 measuring comprehensive performance.

5 Automatic Adaptation to Queries

As mentioned in Section 2, we used the description and title sections of each TREC query as surrogates for the long and short versions of a query respectively. We determined upper bounds on performance on the 249 queries from the TREC 2004 Robust track to provide a target for techniques designed for automatic adaptation to queries.

[4] `http://www.lemurproject.org`

System	MAP	GMAP
Baseline	0.235	0.123
PRF	0.255 (+8.5%)	0.144
Best Sub-query	0.332 (+40.9%)	0.234

Table 3: The utility of sub-query adaptation. The baseline used the *description* portion of the TREC query.

System	MAP	GMAP
Baseline (PRF)	0.261	0.130
Best Expansion Subset	0.341 (+30.76%)	0.218

Table 4: The utility of expansion subset adaptation. The baseline is the PRF-expanded version of the TREC query *title*.

5.1 Upper Bounds

To get a sense of the potential utility of query relaxation for long queries and query expansion for short ones, we performed experiments to obtain upper bounds on performance. For query relaxation we ran retrieval experiments with every combination of query terms from the description portion of 249 Robust 2004 queries. For a query of length n, there are an exponential (2^n) number of combinations. For computational reasons we limited our experiments to combinations of length $n \leq 12$. Table 3 provides an overview of the experimental results. Baseline refers to the situation where we used the description portion of the TREC query without any changes. PRF results in a small improvement over the baseline, i.e. from 0.235 to 0.255. Using the best sub-query of each query (Best Sub-query) resulted in an upper bound in performance almost 40% better than the baseline in terms of MAP, and 100% in GMAP.

Similar experiments for query expansion were performed. Our baseline was the PRF-expanded title portion of each TREC query. PRF was performed using the top 25 documents with the number of expansion terms set to one hundred. To obtain the upper bound (Best Expansion Subset), we considered all subsets of size ten from the top twenty-five terms[5] suggested by PRF. In related work, (Magennis and van Rijsbergen, 1997) performed experiments by considering all subsets of terms from the top twenty suggested by an automatic query expansion technique. We can observe that selecting the best subset for each query resulted in a MAP improvement of 30% and a GMAP improvement of 100%.

As we will notice for other collections too, the potential for improvement in query expansion is less when compared to relaxation. We hypothesize that the reason could be the fact that PRF already provides a competitive starting point.

5.2 Automatic Adaptation

To evaluate the automatic sub-query or expansion subset selection procedures developed in Section 3 we performed retrieval experiments using the queries selected using them. The results for query relaxation, which are presented in Table 6, show that the automatic sub-query selection process was just as good as the baseline. 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. Even in the case of query expansion, where we used only the MaST technique, the resulting performance was only 2% better than the baseline.

The limited utility of the automatic techniques could be attributed to the fact that we were working with the assumption that a technique 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

[5] An ideal experiment would have involved considering all subsets of the one hundred terms, but the exponential number of subsets forced this approximation

	MAP	GMAP
Baseline	0.243	0.136
Average	0.172	0.025
MaxST	0.172	0.025
NE_Average	0.170	0.023
NE_MaxST	0.182	0.029

Table 5: Performance when the highest ranked sub-query selected by various techniques was used. Results are based on a sample of 150 queries from Robust 2004

	MAP	GMAP
Baseline	0.261	0.130
MaxST	0.268	0.136

Table 6: Performance when the highest ranked expansion subset selected using the MaST technique was used.

selected the best-performing sub-query from the top ten ranked by each selection procedure (Table 7). While the effectiveness in each case as measured by MAP is not close to the best possible MAP, 0.332, they are all significantly better than the baseline of 0.243. Similarly, in Table 8 we notice a 10% improvement for query expansion.

6 Interactive Adaptation to Queries

The final results we presented in the last section hinted at a potential for user interaction in the form of IQR and IQE. We envisioned providing the user with a list of the top ten choices made using a good ranking procedure, and asking her to select the option she felt was most appropriate. This additional round of human intervention could potentially compensate for the inability of the selection techniques to select the best sub-query or expansion subset automatically.

6.1 User interface design

Figure 1 is a screenshot of the interface we provided to annotators to guide the system's adaptation to queries.

For IQR, we displayed the description (the *long* query) and narrative portion of each TREC query in the interface. The narrative was provided to help the participant understand what information the user who issued the query was interested in. 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 candidate sub-queries or expansion subsets 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 sub-query from the list. In situations where users observed multiple options

	MAP	GMAP
Baseline	0.243	0.136
AverageTop10	0.296 ($+21.8\%$)	0.167
MaxSTTop10	0.293 ($+20.5\%$)	0.150
NE_AverageTop10	0.278 ($+14.4\%$)	0.156
NE_MaxSTTop10	0.286 ($+17.6\%$)	0.159

Table 7: Performance when the best sub-query from the top ten selected by various techniques was used.

	MAP	GMAP
Baseline	0.261	0.130
MaxSTTop10	0.289 (+10.7%)	0.158

Table 8: Performance when the best expansion subset from the top ten selected by the MaST technique was used.

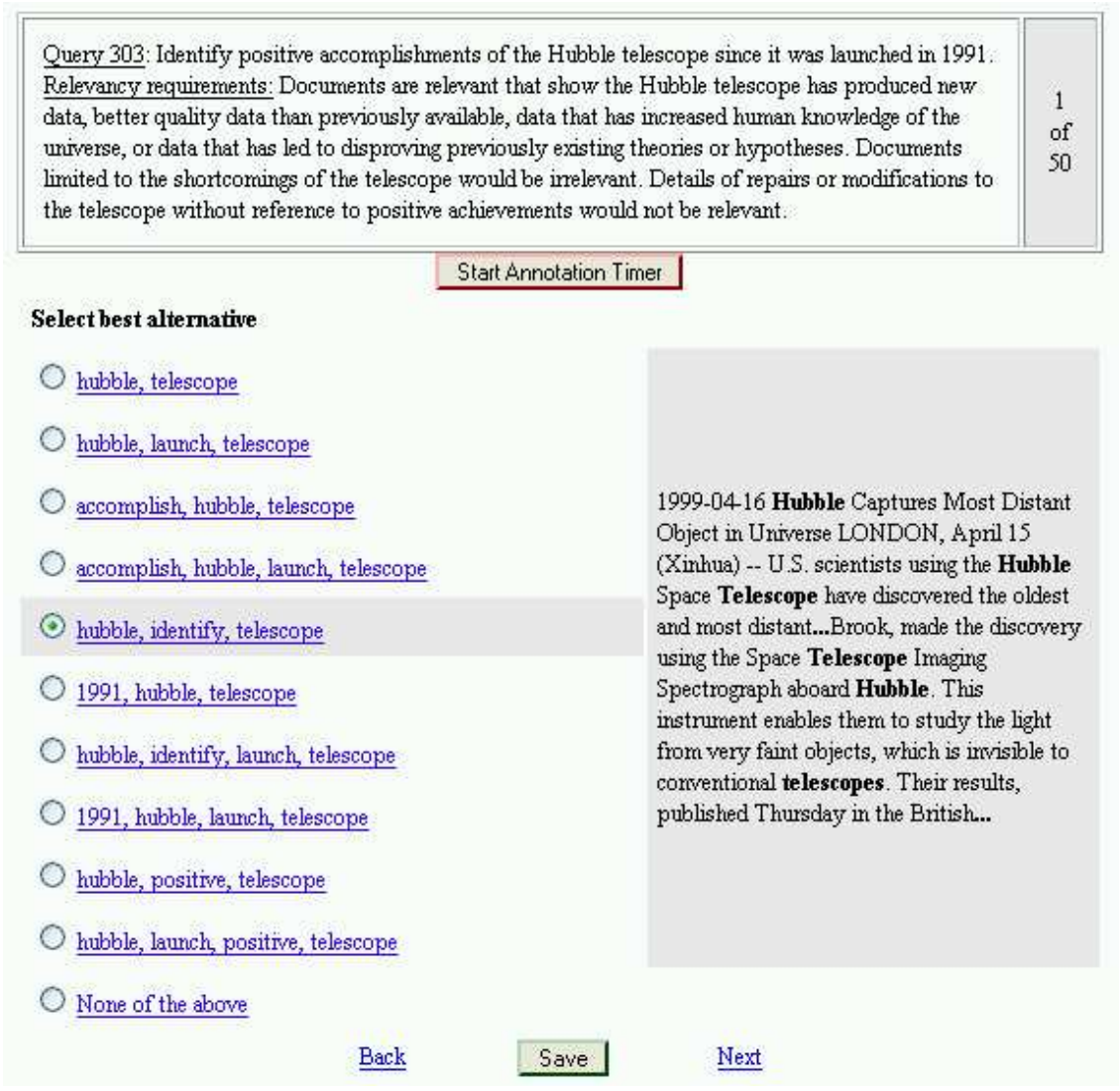

Figure 1: Screenshot of the annotation interface. This particular example is for query relaxation.

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

Table 9: Number of candidates from top 10 that exceeded the baseline

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

Table 10: Results showing the MAP over 19 of 30 queries that the user provided selections for using each context type.

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.

6.2 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 9 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 MaST.

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 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 10). 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 we finally measure MAP only with respect to document ranking, and the snippet was generated from the top-ranked document, we hypothesize that this led to the snippet being a better context to display.

6.3 User Study

We conducted an exploratory user 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

relaxation 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 for this study. For query relaxation, 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.

6.4 Query Relaxation

Table 11 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 utilized 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.

6.5 Query Expansion

Table 12 summarizes the results of our study to evaluate the utility of IQE. While we notice that 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 the relatively little room for improvement (see *Upper Bound* in each case), we believe it is encouraging 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 hope to use this information to reduce the number of options presented to users as part of future work.

7 Efficiency of Interactive Adaptation

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. In such cases it might even be preferable to proceed with the original query. Thus, there is clearly a need to develop a procedure to determine the utility of user-guided adaptation. 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.

Figure 2 sheds light on an interesting aspect of the potential for interactive adaptation to long queries. It shows the distribution of the *absolute* potential improvements in performance through user interaction across the 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 reported in Table 3 masked the minuscule improvements contributed by these queries.

	# 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	10	26	NA	Baseline	0.203	0.159	0. 476	0.507
				With Interaction	0.249	0.199	0.615	0.580
				Upper Bound	0.336	0.282	0.784	0.719
11	11	19	NA	Baseline	0.224	0.156	0.484	0.526
				With Interaction	0.277	0.209	0.652	0.621
				Upper Bound	0.359	0.293	0.810	0.742
12	12	53	NA	Baseline	0.217	0.126	0.452	0.432
				With Interaction	0.276	0.166	0.573	0.501
				Upper Bound	0.354	0.263	0.762	0.654

Table 11: 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. We do not report average time per query for User 6 as the participant overlooked initiating the timer. The results for Users 10, 11 and 12 are from an earlier annotation run where we overlooked measuring the time it took for the annotators to complete each task.

	# 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	3	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 12: 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. All scores, including upper bounds, were calculated only considering the queries for which the participant selected a sub-query.

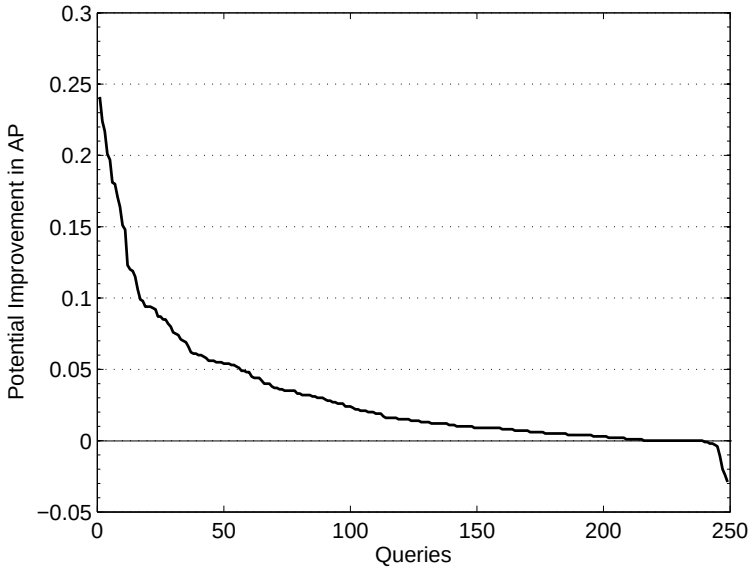

Figure 2: Query Relaxation: The utility of interaction on a per-query basis. Values less than zero (to the right) indicate that none of the sub-queries presented to the user were better than the baseline query

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 relaxed version of the same query?*

Figure 3 provides a pictorial description of the potential utility of user interaction with respect to query expansion. Analogous to query relaxation, 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 following sections we will present techniques we developed towards this goal. We will present the results of simulated user studies. We believe that such studies will enable abstracting away the effects of interface design, experimental methodology, subject experience 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.

8 Adapting to Potential Interaction Utility

There is a large body of previous and related work on procedures to determine the quality of queries (Zhou and Croft, 2006; Cronen-Townsend et al., 2002; Carmel et al., 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 MAP. 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 improvements in MAP, and not just MAP values themselves.

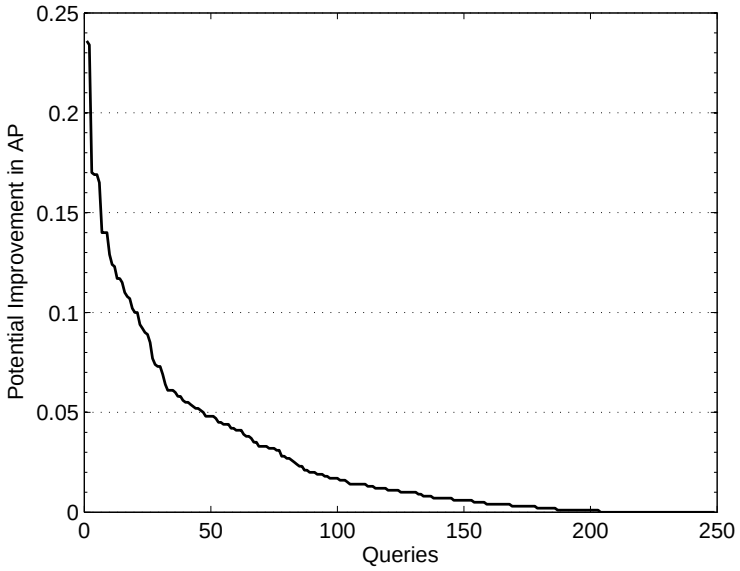

Figure 3: Query Expansion: The utility of interaction on a per-query basis.

8.1 Predictive features

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[6]. Thus, options that have low average length, or are derived from shorter queries, are potentially better

2. The average MaST 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.

8.1.1 Query Relaxation

For each query, we started with the top ten sub-queries ranked by NE_MaxST. 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. In this paper we report only those features that had a high coefficient of correlation with MAP.

Table 13 provides a list of the top four features we found correlating with potential improvements in AP. The γ values in the table are the coefficients of correlation. In this paper, we experiment with using only the first two.

The feature with the highest correlation was original query length (QL). The negative value 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

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

Feature	γ
Original query length	-0.305
Coeff. of Variation	0.245
Mean score	-0.239
Median score	-0.236

Table 13: Features with the highest correlation coefficient with respect to potential improvement in AP

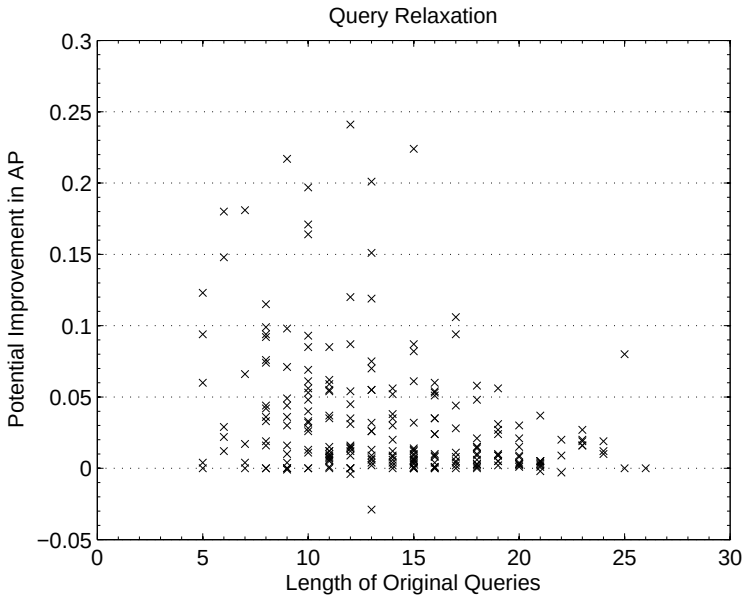

Figure 4: Scatter plot of original query size versus potential improvement in AP due to user interaction

procedure. Figure 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.

The feature with the second highest correlation was a dimensionless quantity, coefficient of variation (CV):.

$$CV = \frac{s_x}{\bar{x}} \tag{5}$$

where s_x is the standard deviation of a set of samples x_i, and $\bar{x}$ its mean. CV can be considered as a measure of the scatter of a set 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 options with varied sub-queries are more likely to cover concepts the user is interested in.

Figure 5 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.

8.1.2 Query Expansion

For each query we considered the top ten expanded queries ranked by MaST. Table 14 lists the features that correlated best with potential improvements.

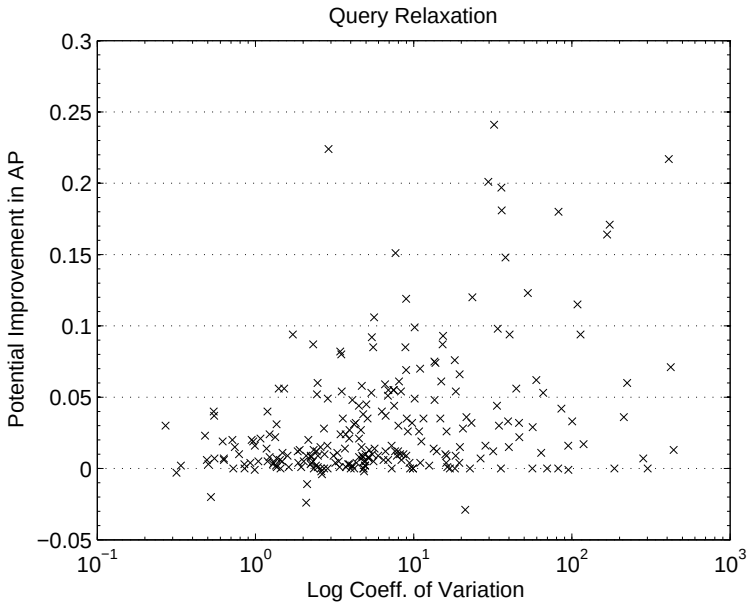

Figure 5: Scatter plot of log coefficient of variation versus potential improvement in AP due to user interaction

Feature	γ
Coeff. of Variation	0.267
Volatility log change	0.171

Table 14: Features with the highest correlation coefficient with respect to potential improvement in AP

		Coeffi cient 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	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 15: Query Relaxation: 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

8.2 Usage Scenarios

We believe there is scope for utilizing predictive measures 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.

8.2.1 System-Centric

Using training instances we learned a decision function to determine when to interact with a user. The high dispersion observed in Figures 4, 5, and corresponding ones for query expansion (not shown) 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 Relaxation

Table 15 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 Wilcoxon signed-rank test, with α set to 0.05.

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 15). For the training set, it meant obviating interaction for 97 i.e. ((1.0-0.61)$*$249) queries in lieu of a 2% reduction in potential MAP improvement . Function 8.1 presents the technique we adopted to determine the utility of interacting with the user in IQR.

Function 8.1: 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

Coeffi cient 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 16: 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

Table 16 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 8.2 presents the technique we developed to determine the utility of interacting with the user in IQE.

Function 8.2: 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 $(decision)$

8.2.2 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 say, 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 apt 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 Relaxation

We utilized the general trend observed in Figure 5 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 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 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.

Query Expansion

We noticed trends similar to Figure 5 for query expansion too, and followed the exact same procedure we adopted in the query relaxation case. Figure 7 depicts similar exploration of the utility of ranking the options by CV before presenting them to the user.

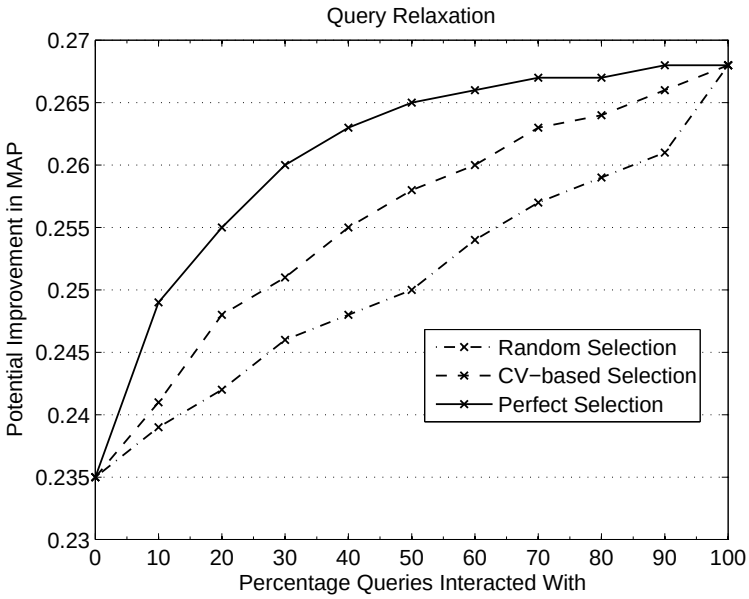

Figure 6: Query Relaxation: Utility of various query ordering procedures when the user places constraints on the number of interactions

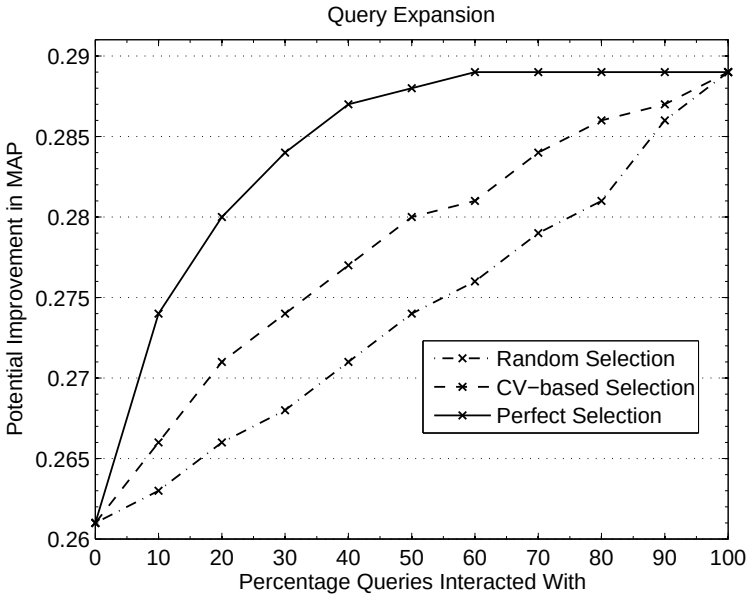

Figure 7: Query Expansion: Utility of various query ordering procedures when the user places constraints on the number of interactions

	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 17: Final results for query relaxation. The reported values are those of MAP

9 Results of Selective Adaptation

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

9.1 Query Relaxation

In Table 17 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 (Wilcoxon 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 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.

9.2 Query Expansion

The results for our experiments with query expansion are given in Table 18. 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. However the MAP achieved by our system was numerically less than that potentially achieved by interacting with all queries.

Figure 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.

For both query relaxation 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 improve-

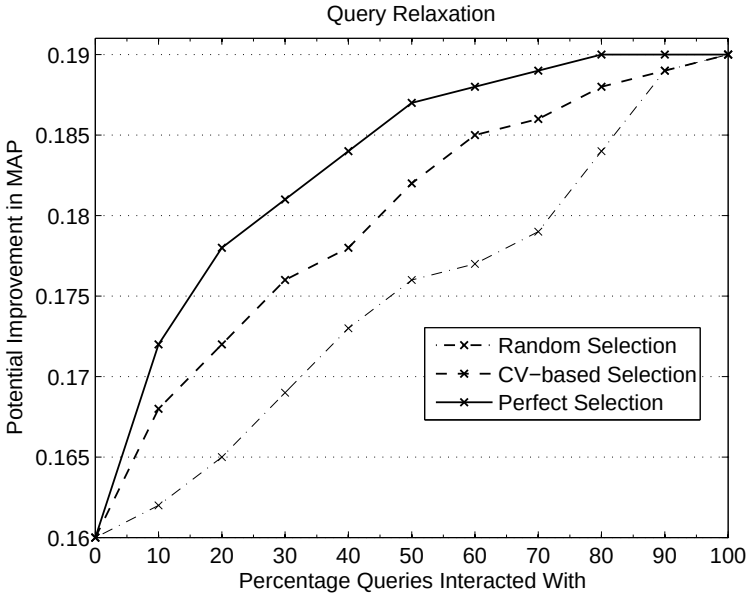

Figure 8: Trajectories of potential improvements in MAP using various question-selection techniques for Robust 2005 in IQR

	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 18: Final results for query expansion. The reported values are those of MAP

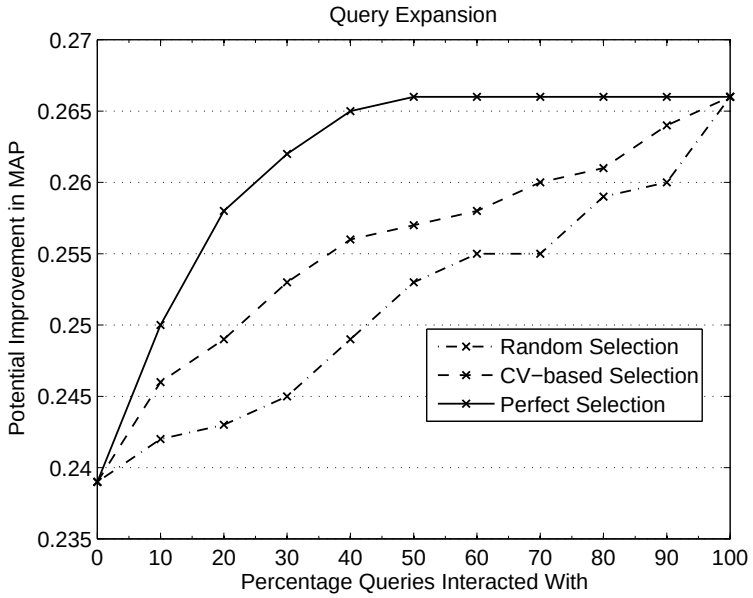

Figure 9: Trajectories of potential improvements in MAP using various question-selection techniques for Robust 2005 in IQE

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

10 Related Work

Our interest in finding a better sub-query or expansion subset that effectively captures the information need is reminiscent of previous work in (Buckley et al., 2000). However, the focus there was more on balancing the effect of query expansion techniques such that different *concepts* in the query were equally benefited. Previous work (Shapiro and Taksa, 2003) in the web environment attempted to convert a user's natural language query into one suited for use with web search engines. However, the focus was on merging the results from using different sub-queries, and not selection of a single sub-query. Our approach of re-writing queries could be compared to query reformulation, wherein a user follows up a query with successive reformulations of the original. In the web environment, studies have shown that most users still enter only one or two queries, and conduct limited query reformulation (Spink et al., 2002). We hypothesize that the techniques we have developed will be well-suited for search engines like Ask Jeeves where 50% of the queries are in question format (Spink and Ozmultu, 2002). More experimentation in the Web domain is required to substantiate this.

Mutual information has been used previously in (Church and Hanks, 1989) to identify collocations of terms for identifying semantic relationships in text. Experiments were confined to bigrams. The use of MaST over a graph of mutual information values to incorporate the most significant dependencies between terms was first noted in (van Rijsbergen, 1979). Extensions can be found in a different field - image processing (Kern et al., 2003) - where multivariate mutual information is frequently used.

Work done by (White et al., 2005) provided a basis for our decision to show context for user studies. The useful result that top-ranked sentences could be used to guide users towards relevant material helped us design an user interface that the participants found very convenient to use. (Borlund and Ingwersen, 1997)'s observations that *simulated work task environments* are equally effective in evaluating the utility of interactive information retrieval systems when compared to actual environments influenced the design of

our user study.

Determining the quality of queries is a continuing challenge, and especially useful for situations like interactive information retrieval. (Cronen-Townsend et al., 2002) developed the clarity measure to serve as a predictive measure for tracking MAP. (He and Ounis, 2004) and (Zhou and Croft, 2007) explored a number of *pre-retrieval* features derived from the query to determine query effectiveness. These include standard deviation of the *idf* scores of the query terms, query length, and query scope. Recent work by (Carmel et al., 2006) attempts to formalize the query difficulty problem.

(Harman, 1988) explored the utility of IQE. Her experiments proved that there was utility in interaction, and users found the guidance provided by the system in the form of terms for expansion useful. (Magennis and van Rijsbergen, 1997) extended these investigations to simulated experiments on a larger scale. The original query was expanded using various sub-sets of predetermined length from the expansion term set. Through exhaustive experiments, the potential of IQE was determined. This is similar to the upper bound experiments we report in Section 5.1. (Ruthven, 2003) extended this idea further by examining various query expansion techniques and performing user studies to compare AQE and IQE. His experiments showed that while there is potential for improvement through IQE, realizing the potential in practice is dependent on a number of limiting factors.

(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.

11 Conclusions

Our results clearly show there is much to be gained by adapting to users' queries. While automatic techniques to do so are not very effective, involving the user in the process clearly helped. We hypothesize that such adaptation 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, helping the IR system 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, the IR system was able to obtain a sense of the users' true information need.

We have also 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.

12 Limitations to our study

We acknowledge several limitations to our study. For instance, the users of our system had to work with pre-specified queries, and not ones that reflected their personal information needs. 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. 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 generalization of our results and conclusions. 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.

13 Future Work

We believe that the work we have presented in this paper could be the starting point for a number of explorations. Better techniques to select effective sub-queries and expansion subsets are important. Identifying parts-of-speech in queries and using them preferentially to construct better query alternatives is a current focus. Analyzing the ranking of documents retrieved by different options presented to the user can help determine not only the quality of the options, but also help determine an optimal number of options. We plan to extend the work on selective adaptation to queries towards learning the ideal adaptive technique (expansion or relaxation) in response to a query. We also intend exploring the adaptation techniques we have proposed in (Kumaran and Allan, 2006).

Acknowledgments

We wish to thank the anonymous reviewers of this manuscript for their very helpful comments. This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023 and in part by NSF Nano grant number DMI-0531171. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

James Allan, James P. Callan, W. Bruce Croft, Lisa Ballesteros, John Broglio, Jinxi Xu, and Hongming Shu. 1996. Inquery at TREC-5. In *TREC*.

James Allan. 2003. The HARD Track Overview in TREC 2003. High Accuracy Retrieval from Documents. In *TREC 12 Proceedings*.

Peter G. Anick and Suresh Tipirneni. 1999. The paraphrase search assistant: terminological feedback for iterative information seeking. In *SIGIR '99: Proceedings of the 322nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 153–159.

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

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

Peter Bruza, Robert McArthur, and Simon Dennis. 1998. Searching the world wide web made easy? The cognitive load imposed by query refi nement mechanisms. In *Proceedings of the Third Australian Document Computing Symposium*, pages 65–71.

Chris Buckley, Mandar Mitra, Janet Walz, and Claire Cardie. 2000. Using clustering and superconcepts within smart: TREC 6. *Information Processing and Management*, 36(1):109–131.

David Carmel, Elad Yom-Tov, Adam Darlow, and Dan Pelleg. 2006. What makes a query diffi cult? In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 390–397.

Kenneth Ward Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *27th ACL Proceedings*, pages 76–83.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2001. *Introduction to Algorithms, Second Edition*. The MIT Electrical Engineering and Computer Science Series. The MIT Press.

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

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 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.

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

D. Harman. 1988. Towards interactive query expansion. In *ACM SIGIR '98*, pages 321–331. ACM Press.

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

Jeffrey P. Kern, Marios Pattichis, and Samuel D. Stearns. 2003. Registration of image cubes using multivariate mutual information. In *Thirty-Seventh Asilomar Conference*, volume 2, pages 1645–1649.

Reiner Kraft, Chi Chao Chang, Farzin Maghoul, and Ravi Kumar. 2006. Searching with context. In *WWW '06: 15th International Conference Proceedings*, pages 477–486.

Giridhar Kumaran and James Allan. 2006. Eliciting information for adaptive retrieval. In *Proceedings of the First International Workshop on Adaptive Information Retrieval (AIR 2006)*, pages 18–19.

Giridhar Kumaran and James Allan. 2007a. A case for shorter queries, and helping users create them. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 220–227, Rochester, NY.

Giridhar Kumaran and James Allan. 2007b. Selective user interaction. In *CIKM '07: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. To Appear.

Victor Lavrenko and W. Bruce Croft. 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.

Mark Magennis and Cornelis J. van Rijsbergen. 1997. The potential and actual effectiveness of interactive query expansion. In *SIGIR '97: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 324–332.

Ragnar Nordlie. 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.

Stephen Robertson. 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.

Ian Ruthven. 2003. Re-examining the potential effectiveness of interactive query expansion. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 213–220.

Jacob Shapiro and Isak Taksa. 2003. Constructing web search queries from the user's information need expressed in a natural language. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 1157–1162.

Xuehua Shen and ChengXiang Zhai. 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 Press.

Amanda Spink and H. Cenk Ozmultu. 2002. Characteristics of question format web queries: An exploratory study. *Information Processing and Management*, 38(4):453–471.

Amanda Spink, Bernard J. Jansen, Dietmar Wolfram, and Tefko Saracevic. 2002. From e-sex to e-commerce: Web search changes. *Computer*, 35(3):107–109.

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

Ellen M. Voorhees and Donna Harman. 1996. Overview of the fi fth text REtrieval conference (TREC 5). In *TREC 5 Proceedings*.

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

Ryen W. White, Joemon M. Jose, and Ian Ruthven. 2005. Using top-ranking sentences to facilitate effective information access: Book reviews. *JAIST*, 56(10):1113–1125.

Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11. ACM Press.

Yun Zhou and W. Bruce Croft. 2006. Ranking robustness: a novel framework to predict query performance. In *CIKM '06: Proceedings of the 15th ACM CIKM Conference*, pages 567–574.

Yun Zhou and W. Bruce Croft. 2007. Query performance prediction in web search environments. In *SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, number 0, pages 543–550, New York, NY. ACM.