A Case for Shorter Queries, and Helping Users Create Them

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

Information retrieval systems are frequently required to handle long queries. Simply using all terms in the query or relying on the underlying retrieval model to appropriately weight terms often leads to ineffective retrieval. We show that rewriting the query to a version that comprises a small subset of appropriate terms from the original query greatly improves effectiveness. Targeting a demonstrated potential improvement of almost 50% on some *difficult* TREC queries and their associated collections, we develop a suite of automatic techniques to re-write queries and study their characteristics. We show that the shortcomings of automatic methods can be ameliorated by some *simple* user interaction, and report results that are on average 25% better than the baseline.

1 Introduction

Query expansion has long been a focus of information retrieval research. Given an arbitrary short query, the goal was to find and include additional related and suitably-weighted terms to the original query to produce a more effective version. In this paper we focus on a complementary problem – *query re-writing*. Given a long query we explore whether there is utility in modifying it to a more concise version such that the original information need is still expressed.

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

Recently, there has been great interest in personalized search (Teevan et al., 2005), where the query is modified based on a user's profile. The profile usually consists of documents previously viewed, web sites recently visited, e-mail correspondence and so on. Common procedures for using this large amount of information usually involve creating huge query vectors with some sort of term-weighting mechanism to favor different portions of the profile.

The queries used in the TREC ad-hoc tracks consist of title, description and narrative sections, of progressively increasing length. The title, of length

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

Proceedings of NAACL HLT 2007, pages 220–227,
Rochester, NY, April 2007.

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. 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.

In the light of the above, we believe there is great utility in creating query-rewriting mechanisms for handling long queries. This paper is organized in the following way. We start with some examples and explore ways by which we can create concise high-quality reformulations of long queries in Section 2. We describe our baseline system in Section 3 and motivate our investigations with experiments in Section 4. Since automatic methods have shortfalls, we present a procedure in Section 5 to involve users in selecting a good shorter query from a small selection of alternatives. We report and discuss the results of this approach in Section 6. Related work is presented in Section 7. We wrap up with conclusions and future directions in Section 8.

2 Selecting sub-queries

Consider the following query:

Define Argentine and British international relations.

When this query was issued to a search engine, the average precision (AP, Section 3) 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. This immense potential for improvement by query re-writing is the motivation for this paper.

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. The query terms were stemmed and stopped.

Analysis of the terms in the sub-queries and the relationship of the sub-queries with the original query revealed a few interesting insights that had potential to be leveraged to aid sub-query selection.

1. Terms in the original query that a human would consider vital in conveying 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.

Given the above empirical observations, we explored a variety of procedures to refine a long query into a shorter one that retained the key terms. We expected the set of terms of a good sub-query to 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.

Some of the sub-query selection methods we explored with these properties in mind are reported below.

2.1 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:

$$I(X;Y) = \sum_x \sum_y p(x,y) log \frac{p(x,y)}{p(x)p(y)} \quad (1)$$

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.

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

$n(x, y)$ 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:

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

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

$$I(X_1; X_2; X_3; ...; X_N) = \quad (4)$$

$$\sum_{i,j=\{1,2,3,...,N; i \neq j\}} I(X_i; X_j) \quad (5)$$

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 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 sub-query selection procedure using the average score as **Average**.

2.2 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 considering another measure that involved creating a maximum spanning tree (MaxST) over the fully connected graph G, and using the weight of the identified tree as a measure representative of the candidate query's quality (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 sub-query selection procedure using the weight of the maximum spanning tree as **MaxST**.

2.3 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 from Section 2, 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 that retained named entities as **NE_Average** and **NE_MasT**.

3 Experimental Setup

We used version 2.3.2 of the Indri search engine, developed as part of the Lemur[2] 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 the 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 and Robust 2005 (Voorhees, 2006) 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. All the documents were from English newswire. We chose these collections because they and their associated queries are known to be *hard*, and hence present a challenging environment. 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*). To determine the best query selection procedure, we analyzed 163 queries from the Robust 2004 track, and used 30 and 50 queries from the 2004 and 2005 Robust tracks respectively for evaluation and user studies.

For all systems, we report mean average precision (MAP) and geometric mean average precision (GMAP). 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 measure comprehensive performance.

4 Experiments

We first ran two baseline experiments to record the quality of the available long query and the shorter version. As mentioned in Section 1, we used the description and title sections of each TREC query as surrogates for the long and short versions respectively of a query. The results are presented in the first two rows, Baseline and Pseudo-relevance Feedback (PRF), of Table 2. Measured in terms of MAP and GMAP (Section3), using just the title results in better performance than using the description. This clearly indicates the existence of terms in the description that while elaborating an information need hurt retrieval performance. The result of using pseudo-relevance feedback (PRF) on both the title and description show moderate gains - a known fact about this particular collection and associated train-

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

		MAP	GMAP
Long Query (Description)	Baseline	0.243	0.136
	PRF	0.270	0.124
Short Query (Title)	Baseline	0.249	0.154
	PRF	0.269	0.148
Best sub-query (Combination)	Baseline	0.342	0.270
	PRF	0.343	0.241

Table 2: Results across 163 training queries on the Robust 2004 collection. Using the best sub-query results in almost 50% improvement over the baseline

ing queries.

To show the potential and utility of query rewriting, we first present results that show the upper bound on performance that can obtained by doing so. We ran retrieval experiments with every combination of query terms. For a query of length n, there are 2^n combinations. We limited our experiments to queries of length $n \leq 12$. Selecting the performance obtained by the best sub-query of each query revealed an upper bound in performance almost 50% better than the baseline (Table 2).

To evaluate the automatic sub-query selection procedures developed in Section 2, we performed retrieval experiments using the sub-queries selected using them. The results, which are presented in Table 3, show that the automatic sub-query selection process was a failure. 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.

The failure of the automatic techniques could be attributed to the fact that we were working with the assumption that 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 selected the best-performing sub-query from the top 10 ranked by each selection procedure (Table 4). While the effectiveness in each case as measured by MAP is not close to the best possible MAP, 0.342, they are all significantly better than the baseline of 0.243.

5 Interacting with the user

The final results we presented in the last section hinted at a potential for user interaction. We envisioned providing the user with a list of the top 10 sub-query candidates using a good ranking procedure, and asking her to select the sub-query she felt was most appropriate. This additional round of human intervention could potentially compensate for the inability of the ranking measures to select the best sub-query automatically.

	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 3: Score of the highest rank sub-query by various measures.

	MAP	GMAP
Baseline	0.243	0.136
AverageTop10	0.296	0.167
MaxSTTop10	0.293	0.150
NE_AverageTop10	0.278	0.156
NE_MaxSTTop10	0.286	0.159

Table 4: Score of the best sub-query in the top 10 ranked by various measures

5.1 User interface design

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. A list of candidate sub-queries was displayed along with links that could be clicked on to display a short section of text in a designated area. The intention was to provide an example of what would potentially be retrieved with a high rank if the candidate sub-query were used. The participant used this information to make two decisions - the perceived quality of each sub-query, and the best sub-query from the list. A facility to indicate that none of the candidates were good was also included.

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

Table 5: Number of candidates from top 10 that exceeded the baseline

5.2 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 greater the number of better candidates in the top 10, the higher the probability that the user would select a better sub-query. Table 5 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.

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 that presenting the passage (Table 6). The reason for this could be that the top-ranking passage we displayed was from a document ranked lower by the document-focussed 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.

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

Table 6: Results showing the MAP over 19 of 30 queries that the user provided selections for using each context type.

6 User Evaluation

We conducted an exploratory study with five participants - four of them were graduate students in computer science while the fifth had a background in the social sciences and was reasonably proficient in the use of computers and internet search engines. The participants worked with 30 queries from Robust 2004, and 50 from Robust 2005[3]. The baseline values reported are automatic runs with the *description* as the query.

Table 7 shows that all five participants[4] were able to choose sub-queries that led to an improvement in performance over the baseline (TREC *title* query only). This improvement is not only on MAP but also on GMAP, indicating that user interaction helped improve a wide spectrum of queries. Most notable were the improvements in P@5 and P@10. This attested to the fact that the interaction technique we explored was 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.

[3]Participant 4 looked that only 34 of the 50 queries presented

[4]The p value for testing statistical significance of MAP improvement for Participant 5 was 0.053 - the result very narrowly missed being statistically significant.

	# Queries	# sub-queries selected	% sub-queries better		MAP	GMAP	P@5	p@10
1	50	26	80.7%	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
2	50	19	78.9%	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
3	80	53	73.5%	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
4	50(34)	19	68.7%	Baseline	0.192	0.142	0.462	0.525
				With Interaction	**0.255**	0.175	0.612	0.600
				Upper Bound	0.344	0.310	0.862	0.800
5	80	65	61.5%	Baseline	0.206	0.111	0.433	0.410
				With Interaction	0.231	0.115	0.486	0.429
				Upper Bound	0.341	0.245	0.738	0.640

Table 7: *# Queries* refers to the number of queries that were presented to the participant while *# sub-queries selected* refers to the number of queries for which the participant chose a sub-query. All scores including upper bounds were calculated only considering the queries for which the participant selected a sub-query. An entry in **bold** means that the improvement in MAP is statistically significant. Statistical significance was measured using a paired t-test, with α set to 0.05.

7 Related Work

Our interest in finding a concise sub-query that effectively captures the information need is reminiscent of previous work in (Buckley et al., 2000). However, the focus was more on balancing the effect of query expansion techniques such that different *concepts* in the query were equally benefited.

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 (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 sub-query selection. 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.

A related problem addressed by (Cronen-Townsend et al., 2002) was determining query quality. This is known to be a very hard problem, and various efforts (Carmel et al., 2006; Vinay et al., 2006) have been made towards formalizing and understanding it.

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.

8 Conclusions

Our results clearly show that shorter reformulations of long queries can greatly impact performance. We believe that our technique has great potential to be used in an adaptive information retrieval environment, where the user starts off with a more general information need and a looser notion of relevance. The initial query can then be made longer to express a most focused information need.

As part of future work, we plan to conduct a more elaborate study with more interaction strategies included. Better techniques to select effective subqueries are also in the pipeline. Since we used mutual information as the basis for most of our subquery selection procedures, we could not consider sub-queries that comprised of a single term. We plan to address this issue too in future work.

9 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor. We also thank the anonymous reviewers for their valuable comments.

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*.

Peter G. Anick and Suresh Tipirneni. 1999. The paraphrase search assistant: terminological feedback for iterative information seeking. In *22nd ACM SIGIR Proceedings*, 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.

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 difficult? In *29th ACM SIGIR Proceedings*, 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.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *25th ACM SIGIR Proceedings*, pages 299–306.

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

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 *15th International CIKM Conference Proceedings*, pages 477–486.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *24th ACM SIGIR Conference Proceedings*, pages 120–127.

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

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.

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.

Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *28th ACM SIGIR Proceedings*, pages 449–456.

Vishwa Vinay, Ingemar J. Cox, Natasa Milic-Frayling, and Ken Wood. 2006. On ranking the effectiveness of searches. In *29th ACM SIGIR Proceedings*, pages 398–404.

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.