

# Exploiting Query History for Document Ranking in Interactive Information Retrieval

Xuehua Shen  
Department of Computer Science  
University of Illinois at Urbana-Champaign

ChengXiang Zhai  
Department of Computer Science  
University of Illinois at Urbana-Champaign

## ABSTRACT

In this poster, we incorporate user query history, as context information, to improve the retrieval performance in interactive retrieval. Experiments using the TREC data show that incorporating such context information indeed consistently improves the retrieval performance in both average precision and precision at 20 documents.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Retrieval models

## General Terms

Algorithms

## Keywords

Query history, query expansion, interactive retrieval, context

## 1. INTRODUCTION

In most existing retrieval models, the retrieval problem is often taken as involving one *single* query and a set of documents. In a real retrieval application (e.g., Web search), the retrieval results using the initial query given by the user may not be satisfactory to the user; often, the user would need to revise the query to improve the retrieval/ranking accuracy [2]. For some information seeking activities, the user may modify his query several times for one information need. In such an interactive retrieval scenario, the information available to us is more than just the current user query and the collection of documents; In general, all the interaction history would be available to us, including the past queries, any possible relevance judgments, and even information about which documents the user has chosen to view. Such history information can potentially be exploited to help improve the retrieval performance for the current query. For example, if the current query is “java”, without knowing any extra information, it would be impossible to know whether it is intended to mean the Java programming language or the kind of coffee called java. As a result, the retrieved documents will likely have both kinds of documents – some may be about the programming language and some may be about the coffee. However, any

particular user is unlikely searching for both types of documents. Such an ambiguity can be resolved by exploiting history information. For example, if we know that the previous query from the user is “cgi programming”, then, it would strongly suggest that it is the programming language that the user is searching for.

In this poster, we explore how to exploit the interaction history in interactive information retrieval. In particular, we study how to improve document ranking for the current query through using the context information about the past queries of the user. We incorporate past queries of the same topic into the current query to obtain a more accurate model of the user’s information need. In some sense, our technique is similar to automatic query expansion [6][3][1], but instead of using pseudo feedback to expand the query, we use the user’s own history queries to expand the current query, which can be combined with pseudo feedback to further improve retrieval performance. Our experiment results show that using the query history consistently improves over using just the current query in both average precision and precision at 20 documents.

## 2. QUERY HISTORY FOR RANKING

In interactive information retrieval, at any moment, we may assume that a sequence of “past queries”, or “query history”,  $q_1, \dots, q_{k-1}$ , are available for a topic, and the current user query is  $q_k$ . Normally, only  $q_k$  is used to rank the documents in the collection. We propose to combine  $q_1, \dots, q_{k-1}$  together with  $q_k$  to have a richer model of the user’s information need. Our basic retrieval model is the KL-divergence retrieval model, which can support model-based feedback [7]. We explore two different strategies – combining query results and combining query models. Combining query results means that we merge the results from  $q_1, \dots, q_k$  by taking an average of the rank of each document. Such a method would reward a document that has been ranked high by all the queries. Combining query models means that we estimate a query language model for each query  $q_i$  using the maximum likelihood estimator, and then take an average of these query models to obtain a context-based query model, which is then used to rank documents with the KL-divergence ranking formula. The probability of a word  $w$  according to the context-based query model is given by

$$p(w|q_1, \dots, q_k) = \frac{1}{k} \sum_{i=1}^k \frac{c(w, q_i)}{|q_i|}$$

where,  $c(w, q_i)$  is the counts of word  $w$  in query  $q_i$  and  $|q_i|$  is the length of query  $q_i$ . Note that this is different from concatenating the query text of  $q_1, \dots, q_k$  to obtain a long query, since we normalize the counts of a query word by the length of each query, which can avoid dominance of one single query. For each document  $d$ , we also estimate a smoothed document language model using the Dirichlet

prior smoothing method [8]. We then rank documents by the KL-divergence value of the query model and the document model. That is, document  $d$  has a score of

$$\sum_w p(w|q_1, \dots, q_k) \log\left(1 + \frac{c(w, d)}{\mu p(w|C)}\right) + \log \frac{\mu}{\mu + |d|}$$

where,  $p(w|C)$  is the collection language model and  $\mu$  is the Dirichlet prior smoothing parameter (set to 2,000 in our experiments).

### 3. EXPERIMENTS

Evaluation of any effect of exploiting query history poses some challenges. First, we need to have a sample of a real user’s query history. Second, we need to have relevance judgments. Web query log is a good source for query history information, but it is very hard to determine the relevant documents for a topic searched by an unknown user. The TREC data have many relevance judgments, but do not have query history information. We decide to exploit the relevance judgments in the TREC data, and generate a sample of query history for some selected topics. Specifically, we built a simple interactive search system using the Lemur toolkit [4], and chose 25 hardest topics, from TREC7 and TREC8 Ad Hoc Tasks[5]. These hardest topics have the lowest best average precision performance given by any system at TREC7 and TREC8. Since a user is unlikely to be satisfied with the initial search results in the case of a hard topic, and is more likely to reformulate queries and browse the retrieved documents many times, using these hardest topics in our experiments approximates well a real world interactive information retrieval scenario.

A Computer Science graduate student is chosen as the subject to use the Lemur search system interactively. For each topic, the subject is told to iteratively improve the query just like what a user typically does when the search results are not satisfactory. He composes the first query using only the topic title. He then reads the top-ranked document returned by the search system, and tries to formulate a better query according to the full TREC topic description and the content of the top document, aiming at improving the ranked list of documents. In this way, the subject generated four versions of queries for each topic. Conceptually, we can think of four time points,  $t_1, \dots, t_4$ , and the subject has a query at each  $t_i$ . All the queries are very short with an average of 3.06 words.

With these data available, we can evaluate the effectiveness of exploiting query history at each time point  $t_i$  by comparing the performance of using the single query  $q_i$  with that of combining  $q_1, \dots, q_i$ . We measure the performance using both average precision and precision at 20 documents (pr@20). The average precision measures the overall ranking accuracy, while pr@20 more reflects the perceived performance by a real user who usually looks at only a few top ranked documents. The results for the query model combination method are shown in the second column and third column of Table 1. We can see that both the average precision and pr@20 are improved by using query history. For example, when we use only the single query  $q_4$ , the average precision and pr@20 are 0.0483 and 0.128 respectively. But when we combine  $q_1, q_2, q_3$  and  $q_4$ , the average precision and pr@20 are 0.0736 and 0.2 respectively, improving 52.4% and 56.3% respectively.

We also combine query history and pseudo feedback to see whether retrieval performance can be further improved by using query history information on top of pseudo feedback. Pseudo feedback is performed on top 3 documents with the mixture model approach [7] and the default parameter values of the Lemur toolkit. The results are shown in the fourth and fifth columns of Table 1. Once again, we see that both the average precision and pr@20 are im-

Query	Simple Retrieval		Pseudo Feedback	
	avg prec	pr@20docs	avg prec	pr@20docs
$q_1$	0.0542	0.154	0.0556	0.152
$q_2$	0.0432	0.136	0.0467	0.144
$q_{1+2}$	0.0569	0.162	0.0572	0.166
Improve.	<b>31.7%</b>	<b>19.1%</b>	<b>22.5%</b>	<b>15.3%</b>
$q_3$	0.0718	0.182	0.0738	0.174
$q_{1+2+3}$	0.0727	0.204	0.0745	0.208
Improve	<b>1.3%</b>	<b>12.1%</b>	<b>1.0%</b>	<b>19.5%</b>
$q_4$	0.0483	0.128	0.0499	0.136
$q_{1+2+3+4}$	0.0736	0.2	0.0754	0.202
Improve	<b>52.4%</b>	<b>56.3%</b>	<b>51.1%</b>	<b>48.5%</b>

**Table 1: Effect of using query history for document ranking.**

proved. Thus the query history really provides extra useful information about the user retrieval activity, which can help a retrieval system to improve the retrieval performance.

The strategy of combining query results has not worked so well, showing only inconsistent improvement in performance. Details are omitted due to the space limit.

### 4. CONCLUSIONS AND FUTURE WORK

In interactive information retrieval, especially for hard topics, the user would generally need to submit a sequence of queries to the search system. We demonstrate that using query history to expand the current query can consistently improve the retrieval performance.

We have only explored the most basic methods for exploiting the query history; more sophisticated analysis is certainly possible and interesting to explore (e.g., term sequence analysis and unequal weighting of different queries). In general, it would be interesting to further study whether and how all the context information, such as the query history, relevance feedback, and documents being edited or viewed by the user, can be used to help the user’s retrieval activity.

### 5. REFERENCES

- [1] W. B. Croft, S. C. Townsend, and V. Larvrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.
- [2] C. Huang, L. Chien, and Y. Oyang. Query session based term suggestion for interactive web search. In *Proceedings of the 10th WWW Conference*, 2001.
- [3] A. S. Mandar Mitra and C. Buckley. Improving automatic query expansion. In *Proceedings of ACM SIGIR 1998*, 1998.
- [4] P. Ogilvie and J. Callan. Experiments using the lemur toolkit. In *Proceedings of the 2001 TREC conference*, 2002.
- [5] E. M. Voorhees and D. K. Harman. Overview of the eighth text retrieval conference (TREC-8). In *Proceedings of TREC-8 Conference*, 2000.
- [6] J. Xu and W. Croft. Query expansion using local and global document analysis. In *Proceedings of ACM SIGIR 1996*, 1996.
- [7] C. Zhai and J. Lafferty. Model-based feedback in the KL-divergence retrieval model. In *Proceedings of CIKM 2001*, 2001.
- [8] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of ACM SIGIR’2001*, 2001.