Simple Questions to Improve Pseudo-Relevance Feedback Results

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

ABSTRACT

We explore interactive methods to further improve the performance of pseudo-relevance feedback. Studies [4] suggest that new methods for tackling difficult queries are required. Our approach is to gather more information about the query from the user by asking her simple questions. The equally simple responses are used to modify the original query. Our experiments using the TREC Robust Track queries show that we can obtain a significant improvement in mean average precision averaging around 5% over pseudo-relevance feedback. This improvement is also spread across more queries compared to ordinary pseudo-relevance feedback, as suggested by geometric mean average precision.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**–Query formulation,Search process

General Terms: Performance, Experimentation

Keywords: User interaction, feedback, information retrieval

1. INTRODUCTION

Improving retrieval performance by automatically or manually reformulating queries [1, 7] is the focus of much research. Approaches that are based on the assumption that top-ranked documents are relevant to the original query can be rendered ineffective if the queries are poorly specified, or if few relevant documents are returned at the top of the ranked list. Some approaches bring the user into the loop by asking her to mark documents at the top of the ranked list as relevant or non-relevant, and use this information for feedback[3]. This could cumbersome to the user. Providing users with an interface to specify the query elaborately and accurately has been tried too. However, such interfaces involve issues ranging from deciding which supplementary information to ask for to the optimal design of the interface.

In this paper we explore a much simpler approach with the goal of improving retrieval performance with minimal participation from the user. We investigate whether asking the user simple questions or requiring the user to make minor modifications to the query will help improve performance to an extent that justifies the little additional effort out in by the user. We acknowledge that there might not be one single approach that improves performance across all queries, but believe that a small number of *simple* questions will take us closer to that goal.

2. EXPERIMENTAL SETUP AND BASELINE

We chose the fifty TREC 2005 Robust Track queries for training, and a set of fifty randomly chosen queries from the TREC 2004 Robust Track as our test set. These queries were tested on the AQUAINT collection, and TREC disks 4&5, minus the Congressional Record, respectively. The choice of Robust Track queries was motivated by the fact that these queries are previously know to be *hard*, and the impact of standard pseudo-relevance feedback is less compared to other query sets. Our baseline queries consisted of terms from the title and description portions of the queries. As our retrieval system, we used version 2.2 of the open-source Indri[1] system. We used the 418 stopwords included in the stop list used by the InQuery system, and the K-stem stemming algorithm implementation provided as part of Indri.

Our baseline system (QL) is a query-likelihood variant of statistical language modeling. The pseudo-relevance feedback mechanism is based on relevance models[5]. For all systems, we report mean average precision (MAP), geometric mean average precision (GMAP), and percentage of queries improved over the QL system. From Table 1 we can observe that PRF improves over QL in both cases, but the improvement across queries is not uniform. The results on the Robust04 queries is even more lopsided, where a gain in MAP is achieved by improving less than a third of the queries, and even showing a drop in GMAP.

3. SIMPLE QUESTIONS

Some measures have been found that loosely correlate with expected gains from PRF [2], but in general there are no obvious ways for deciding when to apply the technique [4].For that reason, we consider some simple questions that could be posed to the user at query time which might reduce the lopsided effect of PRF.

SIGIR'06, August 6–10, 2006, Seattle, Washington, USA.
ACM 1-59593-369-7/06/0008.

[1] `http://www.lemurproject.org/indri`

		QL	PRF	Groups	Patterns	Phrases	Groups + Patterns	Groups + Phrases
Robust05	MAP	0.2278	**0.2750**	**0.3027**	0.2563	0.2307	0.2920	**0.3269**
	GMAP	0.1533	0.1541	0.1777	0.1633	0.1552	0.1563	0.1978
	Queries Improved	–	54%	68%	56%	14%	60%	82%
Robust04	MAP	0.3418	**0.3622**	0.3695	0.3375	0.3508	0.3483	0.3770
	GMAP	0.2155	0.2079	0.2451	0.2266	0.2192	0.2369	0.2541
	Queries Improved	–	29%	33%	21%	11%	28%	32%

Table 1: Performance of the different systems in terms of MAP, GMAP, and percentage of queries improved. Entries in bold face are statistically significant improvements (paired t-test,α = 0.05) . Systems in the last five columns were compared to PRF

3.1 Identifying Topics

Often ambiguous queries like *salsa* retrieve documents from a variety of topics, with deleterious effects on automatic techniques like PRF. One workaround would be to ask the user to select related terms/topics from a list, and then refine the query. Creating such a list on the fly can be a difficult task[1]. Our approach is to utilize a human-generated list of topics - the names of Usenet NewsGroups, and ask the user to select the list(s) they would expect to find their query discussed in. This system is refered to as Groups.

In response to each query, we searched through a Usenet archive spanning twenty years, and returned the titles of the newsgroups occurring in the top two hundred results. Once the user selected the titles, we restricted the query to the particular group(s), and used the results as a topic model for the query. Using this topic model, as well as the collection model, we performed PRF.

3.1.1 Identifying Useful Bigrams

Certain term patterns occur frequently in particular topics. For example, in news reports on *bomb attacks*, a discerning reader can observe that the terms *killed* and *injured* occur frequently within a window of around eight terms. By processing the documents returned from the Usenet groups chosen by the user for each query, we identified the top ten most frequently appearing bigrams. These bigrams, along with the average distance between term interpreted as term window constraints, were appended to the original query, and the resulting system referred to as Patterns.

3.2 Identifying Phrases

The use of phrases to improve retrieval performance is well known. Instead of trying to automatically identify useful phrases, we asked the user to specify the phrases in the query[2]. This system is refered to as Phrases.

4. DISCUSSION AND FUTURE WORK

An overview of the experimental results is provided in Table 1. We can observe that using Usenet group information improves over PRF in both collections. The improvement in GMAP indicates that the Groups system succeeded in causing an improvement across many more queries than PRF. While Patterns and Phrases had better GMAP scores than PRF, their MAP scores were worse. This still translated to better gains when the systems are combined. The system Groups + Patterns betters PRF in terms of GMAP. The best performing system is Groups + Phrases, with better MAP than PRF, and superior GMAP indicating a more balanced improvement across all queries.

Thus we observe that simple inputs from the user can help improve performance beyond the state-of-the-art. This improvement can be obtained by not just doing better on easier queries, but by doing better on a larger set.

We have plans to experiment with other questions like asking the user to add a few terms from the narrative, and asking the user to specify if they were looking for names, locations, organizations or dates. In the former system, by restricting the user to choose from the narrative, which can be treated as a model of what the user has in mind while issuing a query, we can avoid the problems discussed in [6]. We also plan to develop more questions to further refine queries, and study the effect of combining the obtained information.

[2]Only around 40% of the queries had identifiable phrases in them

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 those of the author(s) and do not necessarily reflect those of the sponsor. We thank Prof. R. Manmatha for his suggestions on the use of Usenet archives.

5. REFERENCES

[1] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *ACM SIGIR '03*, pages 88–95, 2003.

[2] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A framework for selective query expansion. In *ACM CIKM '04*, pages 236–237, 2004.

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

[4] D. Harman and C. Buckley. Reliable information access final workshop report. 2003.

[5] V. Lavrenko and W. B. Croft. Relevance based language models. In *ACM SIGIR '01*, pages 120–127, 2001.

[6] I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In *ACM SIGIR '03*, pages 213–220, 2003.

[7] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *ACM SIGIR '96*, pages 4–11, 1996.