York University at TREC 2006: Enterprise Email Discussion Search

Yu Fan[1] *Xiangji Huang*[2] *Aijun An*[1]
iamfanyu@gmail.com jhuang@yorku.ca aan@cse.yorku.ca

[1] *Department of Computer Science & Engineering*
York University, Toronto, ON, Canada

[2] *School of Information Technology*
York University, Toronto, ON, Canada

Abstract

We use the Okapi retrieval system to conduct the email discussion search. The following issues are investigated. First, we make use of the thread structure in the emails to re-rank the documents retrieved by Okapi. We would like to see whether such post-processing of the retrieval result can boost the retrieval performance. Second, in terms of query formulation, we investigate whether the use of only title in a topic achieves better or worse results than the inclusion of other fields such as description and narrative. Third, we investigate whether stemming and stop word removal play an important role in the email search. Our conclusion includes that (1) re-ranking documents using a straightforward method that considers the thread structure can make a small improvement to the retrieval performance, (2) formulating the query using all the fields in a topic achieves the best result, and (3) the use of stemming and stop word removal can improve the performance, but the degree of improvement depends on the stemming method and the stop word list used.

1 Introduction

This is our first time to participate in the TREC Enterprise Track. We focus on the email discussion search task this year. The objective of an email discussion search is to search for messages that contain pro and con in an argument/discussion regarding to a topic. A feature of a collection of emails is that there is a so-called thread structure in the email collection. The thread structure contains a number of trees, where each tree is related to a subject topic and contains all the emails (as tree nodes) connected through "reply by—reply to" relations.

Our primary goals of participating in the discussion search are as follows:

1. Explore the effectiveness of the Okapi system on enterprise search. More specifically, we would like to investigate the effectiveness of Okapi on email discussion search.

2. Investigate the effectiveness of some common techniques in IR on the email data set. In particular, we would like to see how stemming and stop words removal affect the retrieval performance.

3. Evaluate a post processing approach that makes use of the thread structure to re-rank the documents retrieved by Okapi.

2 Our Methods

2.1 Okapi

Okapi is one of the systems that implement a probabilistic retrieval model [1]. The term weighting function used in Okapi is BM25. Given a term t, query q, and document d in a collection of documents, the weight w of d with respect to q and t is calculated by the following formula:

$$w = \frac{(k_1+1)\times tf}{K+tf} \times log\frac{N-n+0.5}{n+0.5} \times \frac{(k_3+1)\times qtf}{k_3+qtf} \quad \oplus \quad k_2 \times nq \times \frac{(advl-dl)}{(advl+dl)} \tag{1}$$

where the variables are described in the following table:

Variable:	description:
N	total number of indexed documents in the collection.
n	number of documents containing the term t. ($n \leq N$)
tf	within document term frequency of t in d.
qtf	within query term frequency of t in q.
nq	number of query term (query length).
dl	length of document d.
avdl	average length of all indexed documents in the collection.
K	$k_1 \times (1-b+b\times \frac{dl}{avdl})$
K1,k2,k3,b	constants, used to tune the system.

In our experiments, we set the parameter values as follows: k1=1.5, k2=0, k3=16 and b=0.55.

2.2 Indexing, Query Processing and Re-ranking

The data set contains 198394 emails posted to 221 email lists. Among them, our parser recognized 159352 true email files. We built indexes on these email files according to different stemming and stop word removal methods. The Okapi system has implemented a strong stemming algorithm and a weak stemming algorithm. A strong stemming algorithm strips off both inflectional suffixes (-s, -es, -ed) and derivational suffixes (-able, -aciousness, -ability), while a weak stemming algorithm strips off only the inflectional suffixes (-s, -es, -ed). We built indexes with no-stemming, weak stemming and strong stemming, respectively, and compare three stemming methods in terms of the retrieval performance. For stop word removal, we also tried three options: no stop word removal, removal according to the sample stop word list provided in Okapi, and removal according to a shorter list that contains 417 single stop terms. The sample stop word list contains single stop terms, a set of two or more terms that can be considered as a phrase (such as "catch up") , some pre-fixes (such as "inter-" and "pre-") and some semi-stop-terms (such as "dec", "1991").

We have three options in query formulation: (1) using only the words in the title of the topic, (2) using the words in the title plus the words in the narrative field, and (3) using

Run	MAP	R-prec	Run Description
0	0.3171	0.3604	No stemming
0R	0.3180	0.3581	Same as run 0, with re-ranking
1	0.3475	0.3885	weak stemming
1R	0.3463	0.3861	Same as run 1, with re-ranking
2	0.3491	0.3848	strong stemming.
2R	0.3492	0.3806	Same as run 2, with re-ranking

Table 1: Stemming function comparison with queries formulated by only titles and the shorter stop word list

the words in all the three fields (title, narrative, description) of the topic. The words that appear in more than 80% of all the topics are removed from the query.

After Okapi outputs a ranked list for a query, we re-rank the documents in the list by using the thread structure. The re-ranking method is straightforward. The top ranked document within a thread is not re-ranked. The other documents will be moved to a position that is closer to the top ranked document in the same thread. There are many ways to determine the new ranks for these documents. We simply move a document to a position that is the average between the document of concern and the top document within the same thread. Our re-ranking method is based on the following assumptions:

1. Okapi can generally find relevant documents.
2. Documents retrieved on the top of the list are more relevant than other documents.
3. Some relevant documents were not well-ranked by Okapi.
4. Documents within the same thread should have similar ranks or at least their ranks are not too far apart.

Note that the re-ranking method has cost. It may drag some of true relevant documents from their top positions to lower ones, and thus negatively affect the performance. The overall effectiveness relies on the soundness of assumptions 1, 3 and 4.

We also tested a blind feedback method, but found that it did not give any improvement in performance. We believe that a well-tuned dictionary would make blind feedback more rewarding.

3 Experimental Results and Discussion

Our goal is to determine the effectiveness of stemming functions, stop word removal methods, query formulation methods and the document re-ranking method on the retrieval performance. Table 1 shows a comparison of stemming functions. For each stemming option, we compare the performance of re-ranking with that of no-re-ranking. We can see that stemming is an effective technique to improve MAP. Between weak and strong stemming methods, the performances are comparable. In terms of MAP, strong stemming is a bit better. But in terms of R-prec, weak stemming is a bit better. We can also see that re-ranking improves MAP slightly, but it decreases R-prec slightly.

Table 2 shows a comparison of query formulation methods. We can observe that using all the fields (title, description and narrative) achieves the best results in both MAP and

Run	MAP	R-prec	Run Description
1(official run york06ed01)	0.3475	0.3885	Title only query, no expansion.
1R	0.3463	0.3861	Same as run 1, with re-ranking
3(official run york06ed02)	0.3312	0.3771	Title expanded with narrative
3R	0.3346	0.3798	Same as run 3, with re-ranking
4(official run york06ed03)	0.3782	0.4195	Title expanded with descriptive and narrative
4R	0.3805	0.4224	Same as run 4, with re-ranking

Table 2: Comparison on query formulation methods with stop words removed with a shorter list and weak stemming used

Run	MAP	R-prec	Run Description
5	0.3588	0.3942	no stop word removal
5R	0.3617	0.3931	Run 5 after re-ranking
6	0.3628	0.3934	removal with the Okapi sample stop word list
6R	0.3630	0.3954	Run 6 after re-ranking
4	0.3782	0.4195	removal with the shorter stop-word list
4R	0.3805	0.4224	Run 4 after re-ranking

Table 3: Comparison on stop word removal methods with weak stemming and title expansed with description and narrative

R-prec, but only expanding the title with the narrative field decreases the performance because of the noise it introduced to the query.

Table 3 compares different stop word removal methods. It shows that removing stop words improves the performance. However, the degree of improvement depends on the stop-word list used. From our experiments, using the stop word list that contains only the common single stop terms is better than using the larger list that contains other types of stop words.

Regarding re-ranking, Tables 2 and 3 show that our re-ranking method can slightly improve the retrieval performance.

Note that among the runs described in Tables 1, 2 and 3, runs 1, 3 and 4 are our official runs submitted to TREC. We submitted 4 runs in total. The run (york06ed04) which is not shown in the tables has errors due to programming bugs.

4 Conclusion and Future Work

Based on our experimental results, we found that re-ranking documents using a straightforward method that consider the thread structure can make a small improvement to the retrieval performance. Also, formulating a query using all the fields in the topic description achieves the best result. However, using title and narrative (without the description field) decreases the performance compared to using only the title field. In addition, we found that the use of stemming improves the performance, and it is the most effective technique to improve the performance among the techniques that we evaluated. Finally, removing stop words can improve the performance as well, but the degree of improvement depends on the stop word list used. It would be interesting to investigate in the future whether a domain-specific stop word list would work better for the discussion search. Another issue that can be explored further is to design and evaluate other re-ranking functions based on the thread structure.

References

1 M.Beaulieu, M.Gatford, X.Huang, S.E.Robertson, S.Walter and P.Williams, Okapi at TREC-5, Proceedings of the 5th Text REtrieval Conference, pp.143-166, 1996.