Experiments on the TREC-8 Filtering Track

Keiichiro Hoashi, Kazunori Matsumoto,
Naomi Inoue, and Kazuo Hashimoto
{hoashi,matsu,inoue,kh}@kddlabs.co.jp
KDD R&D Laboratories, Inc.
2-1-15 Ohara Kamifukuoka, Saitama 356-8502, Japan

1 Introduction

For this year's TREC, KDD R&D Laboratories focused on the adaptive filtering experiments of the Filtering Track. The main focus of our research was the development and evaluation of the profile updating algorithm.

Our profile updating algorithm is based on the query expansion method based on *word contribution*[1][2]. Given manual feedback, our QE method has achieved high performance in the ad hoc track. Therefore, our hypothesis is that this method should work well in the filtering track. We will describe how we implemented this method to the filtering track, and analyze experiments.

Our officially submitted results to TREC were generated from a system with a major bug. The results described in this notebook paper are based on the bug-fixed version of our system.

2 Profile updating

As mentioned in Section 1, the query expansion method based on word contribution was implemented for the filtering track. However, some adjustments were necessary for this implementation. In this section, we will explain the basic idea of this method, describe the adjustments made for the filtering track, and present experiment results.

2.1 Definition of word contribution

Word contribution is a measure which expresses the influence of a word (or term) to the similarity between the query and a document. This is defined by the following formula:

$$Cont(w,q,d) = Sim(q,d) - Sim(q'(w),d'(w)) \quad (1)$$

where $Cont(w,q,d)$ is the contribution of the word w in the similarity between query q and document d, $Sim(q,d)$ is the similarity between q and d, $q'(w)$ is query q excluding word w, and $d'(w)$ is document d excluding word w. In other words, the contribution of word w is the difference between the similarity of q and d, and the similarity of q and d when word w is assumed to be nonexistent in both data. Therefore, there are words which have positive contribution, and words which have negative contribution. Words with positive contribution raise similarity, and words with negative contribution lower similarity.

2.2 Analysis of word contribution

Figure 1 illustrates the contribution of all words from a query and a document relevant to it. The data is sorted in descending order according to the contribution of each word.

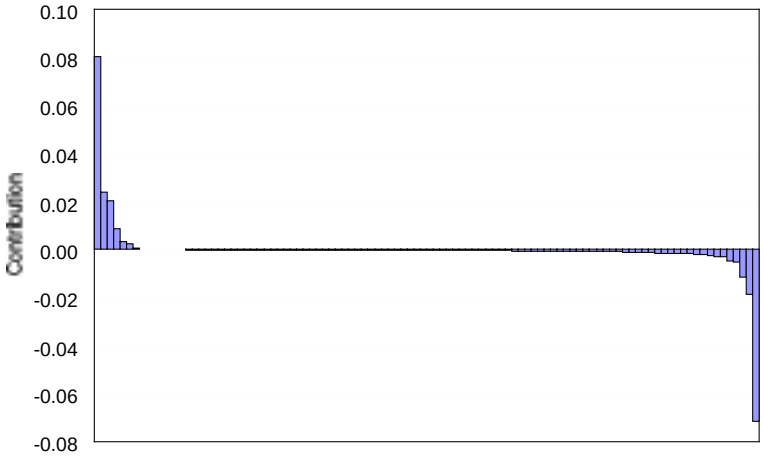

Figure 1: Word contribution between Topic 313 and FBIS3-30043

From Figure 1, it is apparent that there are only a small number of words with highly positive contribution, and a small number of words with highly negative contribution. On the contrary, most words have a contribution near zero, meaning most words do not have a significant influence on the query-document similarity.

As obvious from the definition of word contribution, words with highly positive contribution are presumed to be words that co-occur in the query and document. Such words can be considered as informative words of document relevance to the query. On the contrary, words with highly negative contribution which do not occur in the original query can be considered as words which discriminate relevant documents from other non-relevant documents contained in the data collection.

Since the main objective of query expansion is to add words which are effective in distinguishing relevant documents from the data collection, we assumed that words with highly negative contribution are extremely suitable for expanding the original query. Moreover, we presumed that value of word contribution is a measure of the importance the word has for discrimination. Therefore, the application of word contribution values as the weight of the extracted word for query expansion should also be effective.

2.3 Query expansion method

Based on our arguments in the previous section, we have developed the following query expansion method.

First, the word contribution of all words in the query and relevant documents are calculated. If there are Num documents which are relevant to the query q, the relevant document set for q is $D_{rel}(q) = \{d_1, \cdots, d_{Num}\}$. From each relevant document d_i, N words with the lowest contribution are extracted.

Next, a score for each extracted word w is calculated by the following formula:

$$Score(w) = wgt \times \sum_{d \in D_{rel}(q)} Cont(w, q, d) \quad (2)$$

where wgt is a parameter with a negative value (since the contribution is also negative). Calculated scores are regarded as the term frequency values of each word. Therefore, when using the TF*IDF method, the IDF of the word is multiplied to this score to get its final weight. Finally, all extracted words and their weights are added to the original query. If any of the extracted words occur in the original query, that word is not added to the new query. Words with negative scores were also excluded from the expanded query.

2.4 Implementation to profile updating

The query expansion method described in the preceding section requires a "set" of relevant documents. However, since documents come into the system one by one in the filtering process, this set of relevant documents cannot be made unless the system accumulates results. We did not apply this method to our system. Instead, we calculated the word contribution of selected words occurring in retrieved documents, and added them to the profile.

Although our query expansion method proved to be effective without the use of information from non-relevant documents, we felt the necessity to use this information for the filtering process. Therefore, we took a Rocchio-like approach[3] to apply non-relevant document information to the profile. First, the weights of each selected word from non-relevant documents were calculated by the same method as with relevant documents. Next, instead of adding the calculated weight, we subtracted it from the original profile. Words with negative weights resulting from this process are not used for similarity calculation, but all weights are preserved in the profile vector. Therefore, words extracted from both relevant and non-relevant documents have smaller weights than words which are only extracted from relevant documents.

2.5 Additional System Details

Our system is based on the vector space model. The weighting calculation scheme is based on the TF*IDF based weighting formulas for the SMART system at TREC-7 [4], with minor customizations. The TF and IDF factors for our system are as the following:

- TF factor
$$\log(1 + tf) \quad (3)$$
- IDF factor
$$\log\left(\frac{M}{df}\right) \quad (4)$$

where tf is the term's frequency in the document, df is the number of documents that contain the term, and M is the total number of documents in the data collection. The document frequency data was generated from TREC CD-ROMs Vol 4 and 5, excluding (of course) the Financial Times documents. We added 1 to the term frequency inside the logarithm of the TF factor because the tf value resulting from word contribution occasionally has values below 1, which results in a negative weight.

Different weights were set for the calculation of scores from word contribution data, based on the relevance of the document the word was extracted from. Hereafter, w_{rel} expresses the weight for words extracted from retrieved relevant documents, and w_{nrel} expresses the weight for words extracted from retrieved non-relevant documents.

Moreover, we did not make use of the controlled-language field of the Financial Times database for our experiments.

3 Profile updating experiments

In this section, we will describe the experiments made for evaluation of the profile updating algorithm.

3.1 Experiment conditions

The main interest of our experiment is the influence of the 2 parameters of our algorithm, w_{rel} and w_{nrel}, to filtering results. In this experiment, we set w_{rel} to {-200, -400, -800}, w_{nrel} to {-100, -200, -400, -800}, and ran all possible parameter combinations (12 runs). For comparison, we also made an experiment run without profile updating.

3.2 Results

Table 1 shows the average scaled utility[5] of each run for all combinations of w_{rel} and w_{nrel}. For comparison, the average scaled utility of the run without profile updating ("$NoPU$") is also shown in this table. The parameter s used for scaled utility calculation is set to 200.

Table 1: Average scaled utility (s=200), FT92-94

	w_{nrel}			
w_{rel}	-100	-200	-400	-800
-200	0.4558	0.4840	0.5091	0.5257
-400	0.4172	0.4777	0.5107	0.5184
-800	0.3815	0.4349	0.4842	0.5100
$NoPU$	0.3807			

As obvious from Table 1, we achieved significant improvement of performance compared to the $NoPU$ run, except when the absolute value of w_{rel} is much higher than that of w_{nrel}, as in the case when $\{w_{rel}, w_{nrel}\} = \{-800, -100\}$. Furthermore, scaled utility constantly improves as the absolute value of w_{nrel} increases.

3.3 Discussions

Although we have achieved overall improvement by applying query expansion based on word contribution to our filtering system, the utility is not satisfactory. For further analysis, we examined the results for each year of Financial Times data (FT92, FT93, FT94). Figures 2 - 4 illustrate the improvement of average scaled utility for each year compared to the $NoPU$ run.

As observed from the analysis of yearly results, the profile seems to improve if sufficient information from relevant documents are fed back to the profile. However, the excessive retrieval of non-relevant documents before sufficient retrieval of relevant documents lowers the total utility, resulting in the total decline of performance.

The main cause of this problem is that our algorithm does not utilize information from non-relevant documents except for the Rocchio-like approach described in Section 2.4. Subtracting the weights of words extracted from non-relevant documents contributes to higher performance by lowering the influence of words which occur in both relevant and non-relevant documents. However, this does not affect the profile until a relevant document has been retrieved, since negative values in the profile vector are not used for similarity calculation.

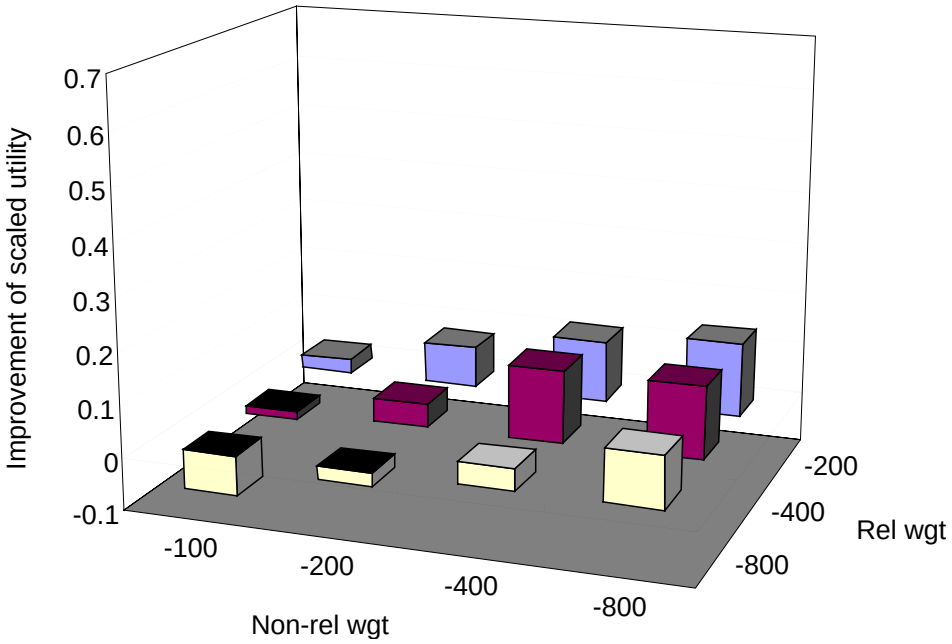

Figure 2: Improvement of scaled utility compared to $NoPU$ (FT92)

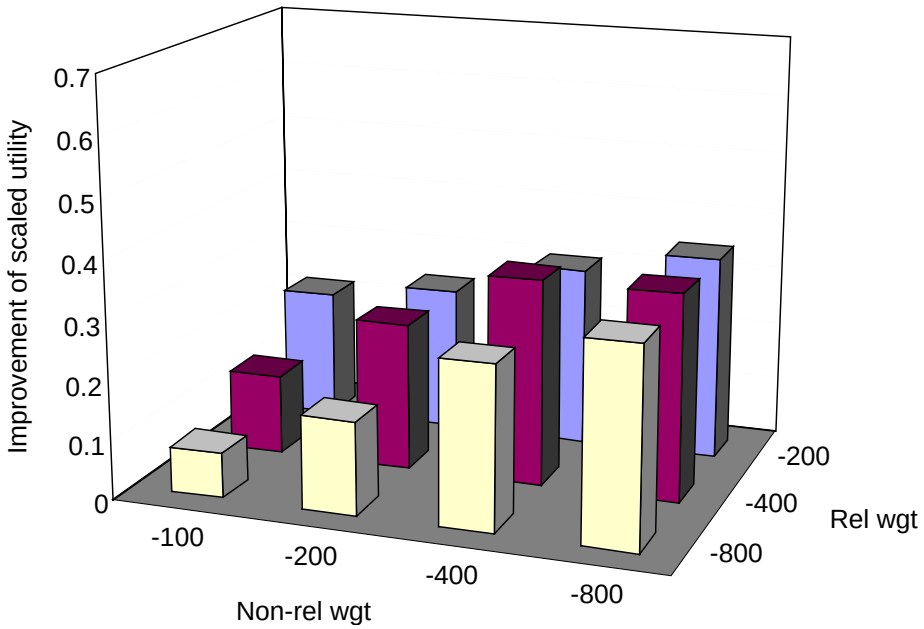

Figure 3: Improvement of scaled utility compared to $NoPU$ (FT93)

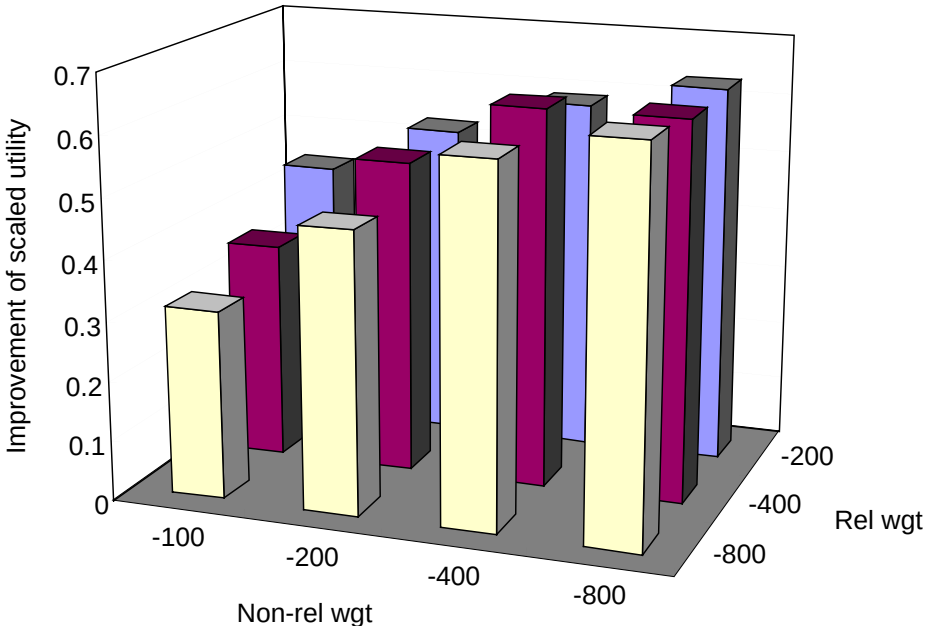

Figure 4: Improvement of scaled utility compared to $NoPU$ (FT94)

Therefore, it is necessary to develop a method which makes more use of information from non-relevant documents. Such a method should decrease the number of mistakenly retrieved documents without affecting the retrieval of relevant documents.

4 Conclusion and Future Work

We have made experiments to evaluate the profile updating algorithm and the threshold adjustment algorithm. Experiments on profile updating showed promising results, although there is a necessity to improve our algorithm to apply more information from mistakenly retrieved non-relevant documents to the profile.

We are currently working on an algorithm which makes the use of a profile which expresses the features of past selected non-relevant documents. We hope to evaluate this method in future TREC filtering experiments.

Acknowledgments

The authors appreciate Akiko Onishi of Waseda University and Rickard Johansson of Uppsala University for their great efforts on the experiments described in this paper. We also appreciate the fellow researchers in the Knowledge-Based Information Processing Lab of KDD R&D Laboratories for their advice.

References

[1] K Hoashi, K Matsumoto, N Inoue, K Hashimoto: "TREC-7 Experiments: Query Expansion Method Based on Word Contribution", The 7th Text REtrieval Conference, NIST SP 500-242, pp 433-441, 1999.

[2] K Hoashi, K Matsumoto, N Inoue, K Hashimoto: "Query Expansion Method Based on Word Contribution", Proceedings of SIGIR'99, pp 303-304, 1999.

[3] J Rocchio: "Relevance Feedback in Information Retrieval", in "The SMART Retrieval System – Experiments in Automatic Document Processing", Prentice Hall Inc., pp 313-323, 1971.

[4] A Singhal, J Choi, D Hindle, D Lewis, and F Pereira: "AT&T at TREC-7", The Seventh Text REtrieval Conference, pp 239-251, 1999.

[5] D Hull: "The TREC-7 Filtering Track: Description and Analysis", The 7th Text REtrieval Conference, NIST SP 500-242, pp 33-56, 1999.