SCAI TREC-8 Experiments

Dong-Ho Shin, Yu-Hwan Kim, Sun Kim,
Jae-Hong Eom, Hyung-Joo Shin, Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, Korea
E-mail: {dhshin, yhkim, skim, jheom, hjshin, btzhang}@scai.snu.ac.kr

ABSTRACT

This working note reports our experiences with TREC-8 on four tracks: Ad Hoc, Filtering, Web, and QA. The Ad Hoc retrieval engine, SCAIR, has been used for the Web and QA experiments, and the filtering experiments were based on it's own engine. As a second entry to TREC, we focused this year on exploring possibilities of applying machine learning techniques to TREC tasks. The Ad Hoc track employed a cluster-based retrieval method where the scoring function used cluster information extracted from a collection of precompiled documents. Filtering was based on naive Bayes learning supported by an EM algorithm. In the Web track, we compared the performance of using link information to that of not using the information. In the QA track, some passage extraction techniques have been tested using the baseline SCAIR retrieval engine.

1 Introduction

In TREC-8, SCAI participated in 4 different tracks: Ad Hoc, Filtering, Web and QA. Among these only the Ad Hoc track is the second entry, others are the first participation. The Ad Hoc, Web, and QA tasks have been based on the SCAI information retrieval engine, SCAIR. Due to the different characteristics, the Filtering experiments were based on its own engine and storage system. After the release of the TREC-7 results, we experimented in various ways to get more experiences. This year our focus has been exploring various possibilities of using machine learning algorithms to improve the performance on TREC tasks.

Our retrieval engine, SCAIR, has been upgraded from the experiences of last year. SCAIR manages documents in inverted file structure and supports some convenient APIs to higher applications. A number of 556 stop words and 335879 indexing terms are used. Porter's stemming algorithm [4] and a modified suffix truncation algorithm are implemented. These are not for indexing but for retrieval, so more flexible retrieval is possible by setting some parameters. WordNet is also embedded to manipulate query terms.

This paper is organized as follows. In Section 2, we describe and report on the Ad Hoc exper-

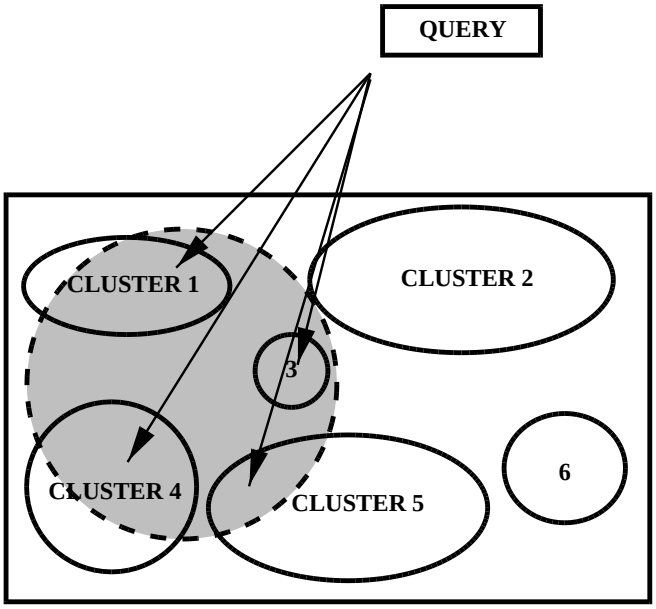

Figure 1: Document clusters and the Ad Hoc retrieval

iments. Section 3 discusses the result of the filtering track based on naive Bayes classifiers trained with expectation maximization (EM). Section 4 analyzes the effect of using link information in the Web track. Section 5 reports on the experimental results of the QA task. Section 6 summarizes our experiments at TREC-8.

2 Ad Hoc Track

Our Ad Hoc retrieval is based on the vector space model. Formally, a document is represented as a list of terms or vectors. A document collection is represented as a term-document matrix which are normally very sparse. A query consists of terms, too.

The documents are indexed by the classical $tf \cdot idf$ weighting scheme:

$$w_{ij} = tf_{ij} \cdot \log\left(\frac{N}{df_j}\right), \tag{1}$$

where w_{ij} is the weight of jth term in the ith document, tf_{ij} is the frequency of the jth term in the ith document, N is the total number of documents in the collection, and df_j is the number of documents in which the jth term occurs. Query terms are weighted only by the idf value.

The similarity between a document and a query is measured by cosine coefficient:

$$S_{ij} = sim(d_i, q_j) = \frac{\sum_{k=1}^{n} w_{ik} \cdot q_{jk}}{\sqrt{\sum_{k=1}^{n} {w_{ik}}^2 \cdot \sum_{k=1}^{n} {q_{jk}}^2}} \tag{2}$$

where w_{ik} and q_{jk} are term weights for document i and query j, respectively. After query-document similarities are measured, an ordered list of documents is produced. Clustering information was then used to rearrange the list.

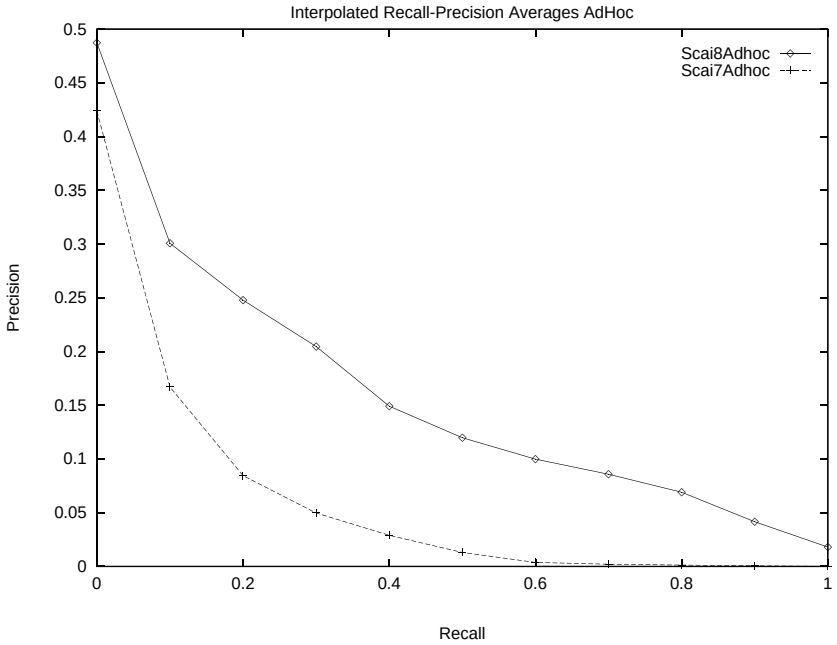

Figure 2: Ad Hoc results

The cluster information was obtained from a precompiled collection of documents. Figure 1 illustrates the relation of clustering and retrieval. The rationale behind this approach is as follows. In general retrieval situations, documents are retrieved from multiple clusters due to query ambiguity. But user's information needs are usually concentrated on one category. Thus, by excluding non-relevant clusters, retrieval performance might be improved.

A hierarchical clustering method is used to partition the whole document collection into a number of disjoint subcollections. Let y_i denote the score of document i. Then, the average score of the original collection of N documents are

$$\bar{Y} = \frac{\sum_{i=1}^{N} y_i}{N}. \tag{3}$$

After clustering, the average score of cluster k to which document i belongs is given as

$$\bar{c}_k^i = \frac{\sum_{i \subseteq \mathcal{C}_k} y_i}{N_k}, \tag{4}$$

where $\mathcal{C}_k$ is the kth cluster and N_k is its size. Using this information, the similarity of document d_i to query q_j is now updated by

$$S'_{ij} = S_{ij} \frac{\bar{c}_k^i}{\bar{Y}} \tag{5}$$

Our assumption is that the top N documents retrieved are relevant, and the clusters which have many relevant documents are considered as relevant clusters. This mechanism gives higher scores to the documents that belong to the relevant clusters, though they are not selected into the top N list. In the experiments we set $N = 100$. This mechanism is similar to 'winner and his neighbor

gang take all' rule. Figure 2 compares our TREC-7 vs. TREC-8 Ad Hoc results. As can be seen in the figure, a significant improvement in Ad Hoc performance was achieved by using the additional cluster information.

3 Filtering Track

We participated in the batch filtering track. As the first entry to the filtering track in TREC, our main concern this year was to explore the possibilities of machine learning algorithms for information filtering. We experimented with naive Bayes classifiers trained with the expectation-maximization algorithm [2].

Naive Bayes is a statistical approach to tackle classification problem. In the Bayesian approach, the most probable target value is assigned to the new document.

$$P(c_k|d_i) = \frac{P(c_k)P(d_i|c_k)}{P(d_i)} = \frac{P(c_k)\prod_{t=1}^{|d_i|} P(w_{it}|c_k)}{P(d_i)} \tag{6}$$

where d_i is the ith document, c_k is the kth class, and t is the index over the terms in d_i. Naive Bayesian classifiers assume that terms are conditionally independent of target value, thus assuming the second equality in the above equation. EM algorithm has been used to improve the performance of this algorithm by using unlabeled documents for getting more reliable statistics. Because TREC data has only small number of labeled documents and almost all the documents have no labels, EM algorithm may be useful in this situation.

We used no special techniques for preprocessing. Stop-words were removed and words were stemmed by the Porter's algorithm. The usual $tf \cdot idf$ was used to weight the terms. Cosine normalization was used. We used only FT92. No other documents, such as FT91, or thesaurus were used.

Figure 3 plots the LF1 values for our experiments. The result is moderate, averaging 4.16. Sometimes, naive Bayes with EM scored much higher than the average of the whole runs except the worst. In topic 352 and topic 389, for example, we achieved highest score, each 149 (followed by 130) and 218 (followed by 176). Usually, they had many positive examples. It can be concluded that naive Bayes with EM can achieve better performance with many positive examples.

4 Small Web Track

In our entry to the small Web track, our concern centered around the following two questions: 1. Do the best methods in the TREC Ad Hoc task also work best on the Web data (WT2g collection), and 2. Can link information in Web data be used to obtain more effective search rankings than can be obtained using page content alone. We first applied the method that was used for the Ad Hoc task to the Web task and then re-ranked its ranking results using link information.

We used a relatively simple re-ranking method. The top-ranking 2000 documents were chosen for re-ranking. We assumed that a document d and the set of documents, D_i, which are linked to d_i belong to the same class. We used only inlink documents (the documents that have links to

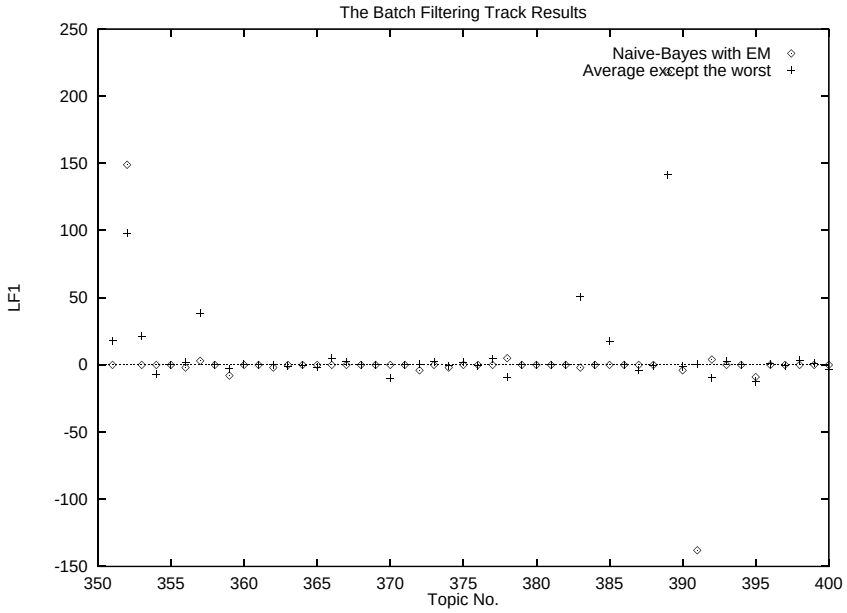

Figure 3: Batch filtering results.

document d_i), not outlink documents (the documents which document d_i has links to). This was motivated by the object-oriented concept where the objects in a higher class contains the objects in lower classes. We tested various re-ranking methods and made some preliminary experiments. In the following we report on a method that was used to get the results we submitted to TREC-8 Small Web track.

Let y_i be the score of document i. Let ℓ be the index of inlink documents of d_i, and y_ℓ its score. Then, the score of document d_i is updated by using the scores of inlink documents as follows:

$$y_i \leftarrow y_i + \alpha \left(\sqrt{\sum_{\ell=1}^{L_i} y_\ell} - y_i \right), \tag{7}$$

where α is a small constant.

Figure 4 shows the result of runs using the re-ranking method described above. The results for using contents only and for contents combined with link information are compared. The result of the Ad Hoc task is also shown. It can be observed that there is no significant difference in using link information for the Web documents. It seems that effective use of link information is not so trivial. An assumption in our use of link information was that the main topic of a document and its inlink documents are the same or very similar. It turns out, however, that the whole links are not always so closely related to each other and thus care must be taken to choose linked documents which are most significant. It can be said that the accuracy of the Ad Hoc engine influences the contribution of link information.

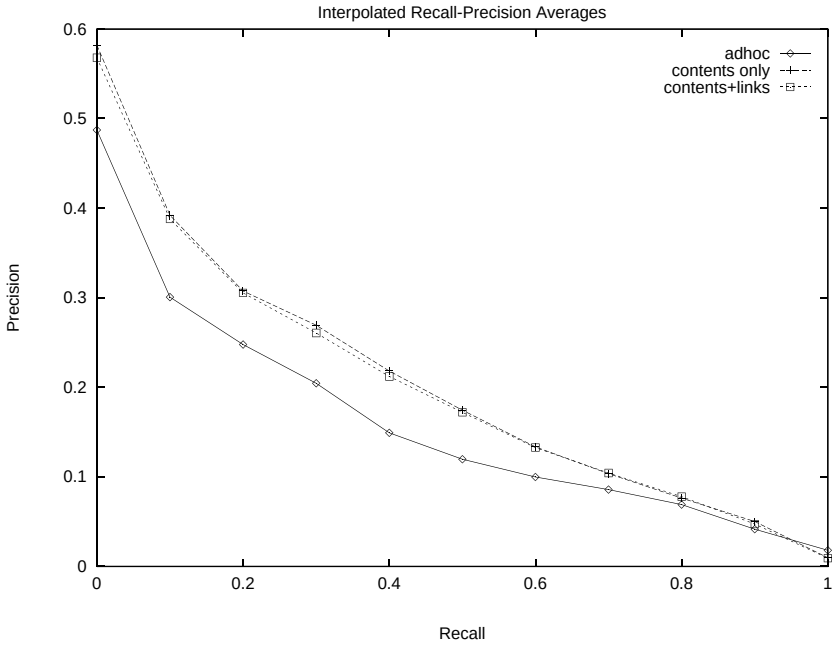

Figure 4: Small Web track results

5 Question Answering Track

The aim of the QA track is to get 'information' not 'documents' as retrieval results. Phrases are extracted for the user query. To achieve this purpose, we used a two-stage retrieval procedure. In the first stage, relevant documents for a user query are retrieved by the adhoc retrieval engine. In the second stage, relevant passages are extracted from those documents by text-snippet extraction.

Text-snippets are extracted by determining local contexts, i.e. the phrases near the position of the highest-weighted query term. We refer to these contexts as answering zones. According to the QA rule, each candidate answering zone is 250 bytes long. More than one answering zones can be constructed within a document. If other important terms co-occur within the answering zone, the candidate zone get a high score. Then, the answering zone with the maximum score is selected as the final answer.

The experimental results were not very surprising, considering the difficulty of the task and our limited experience in this domain. We obtained 44 correct answers out of 198 questions. By correct we mean the answers were among the top-five candidate answers officially provided by TREC-8 QA track. About half of the correct answers were within top-two answers. Figure 5 analyzes the difference of our score from the median value for the 25 QA runs (entries) on each question. The difference of 1 for the questions in Figure 5 means that the QA runs from other entries found the correct answer whereas ours did not make it. The value of 0 indicates that our system achieved better results than other entries. Intermediate values indicate the relative degree of goodness of our results compared with other entries to QA task this year. It seems that the quality of QA accuracy seems very much dependent on the quality of the adhoc retrival engine.

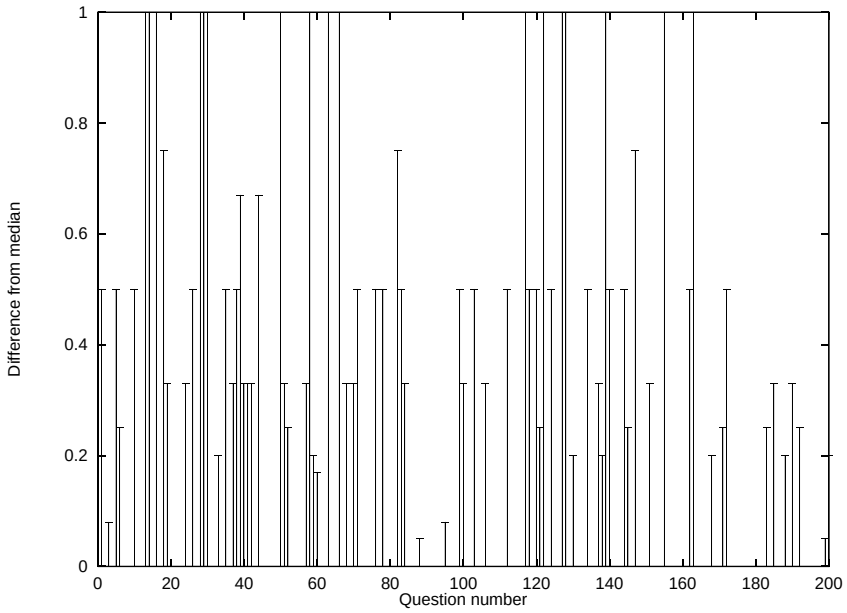

Figure 5: QA track results

6 Conclusions

In the Ad Hoc task, we achieved a significant performance improvement over our results obtained last year. It seems that the use of cluster information in the retrieved documents, in addition to the document-query similarity, is useful for enhancing precision of retrieval. In filtering, we tested the possibility of naive Bayes learning and achieved improved performance by using an EM algorithm, though the absolute performance was not very satisfactory. Our limited experiments in the small Web track showed the importance of a proper use of link information. The QA track performance was not satisfactory but our approach based on 'answering zones' seems promising.

Acknowledgements

This research was supported in part by the Korea Ministry of Information and Telecommunications under Grant C1-98-0068-00 through IITA.

References

[1] Boyan, J., Freitag, D. and Joachims, T., A Machine Learning Architecture for Optimizing Web Search Engines, *AAAI-96 Workshop on Internet-based Information Systems*, 1996.

[2] McCallum, A., Nigam, K., and Thrun, S., Learning to Classify Text from Labeled and Unlabeled Documents, *Machine Learning*, 1999.

[3] Picard, J., Modeling and Combining Evidence Provided by Document Relationships Using Probablistic Argumentation Systems, *SIGIR-98*, pp. 182-189, 1998.

[4] Porter, M.F., An Algorithm for Suffix Stripping, *Program*, 14(3), pp. 130-137, 1980.

[5] Robertson, S.E. and Sparck Jones, K., Relevance Weighting of Search Terms. *Journal of the American Society for Information Science* 27, 1976.

[6] Rocchio, J., Relevance Feedback Information Retrieval, In G. Salton, editor, *The Smart Retrieval System - Experiments in Automatic Document Processing*, Prentice-Hall, pp. 313-323, 1971.

[7] Silverstein, C. and Pedersen, J.O., Almost-Constant-Time Clustering of Arbitrary Corpus Subsets, *SIGIR-97*, pp. 60-66, 1997.

[8] Schütze, H. and Silverstein, C., A Comparison of Projections for Efficient Document Clustering, *SIGIR-97*, pp. 74-81, 1997.

[9] Yang, Y., Noise Reduction In A Statistical Approach To Text Categorization, *SIGIR-95*, 1995.