A Two-Stage Retrieval Model for the TREC-7 Ad Hoc Task

Dong-Ho Shin
Artificial Intelligence Lab (SCAI)
Interdisciplinary Program in Cognitive Science
Seoul National University
Seoul 151-742, Korea
E-mail: dhshin@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, Korea
E-mail: btzhang@scai.snu.ac.kr

ABSTRACT

A two-stage model for ad hoc text retrieval is proposed in which recall and precision are maximized sequentially. The first stage employs query expansion methods using WordNet and on a modified stemming algorithm. The second stage incorporates a term proximity-based scoring function and a prototype-based reranking method. The effectiveness of the two-stage retrieval model is tested on the TREC-7 ad hoc text data.

1 Introduction

Performance of text retrieval systems is usually measured on the basis of recall and precision. Recall is defined as the proportion of relevant documents retrieved, while precision is the proportion of retrieved documents that is relevant. We wish both high recall and high precision, but the one should usually be traded for the other. To increase recall, just retrieving many documents would be helpful. But then precision decreases and vice versa. In our early experiments on TREC collections we tried to improve retrieval performance by directly optimizing a combination of both factors. However, managing so large a size of documents in one homogeneous model was both inefficient and ineffective. In addition, many techniques were not applicable simply due to its computational efforts for optimizing both factors at the same time. These failures led us to use a two-stage model which deals with recall and precision separately.

In the two-stage retrieval model, we first attempt to maximize the recall performance and then try to improve precision subsequently. One advantage of this approach is that the effectiveness of several techniques can be analyzed separately since this separation reduces the interference effect of recall and precision. Another advantage is that this separate optimization can reduce computational overhead since the techniques for improving precision are applied to a small subset of documents which have been retrieved in the first stage. In this article, we describe the techniques for improving recall and precision separately, and report the experimental results obtained on the TREC-7 ad hoc task.

The paper is organized as follows. In Section 2, we describe the system architecture of the two-stage retrieval model. Section 3 describes the techniques we studied for improving recall and precision. Section 4 reports the experimental results on the TREC-7 ad hoc document set using combinations of the techniques in the framework of the two-stage retrieval model. Section 5 draws our conclusions from these experiments.

2 The Two-Stage Retrieval Model

Figure 1 illustrates the architecture of our information retrieval system for the TREC-7 ad hoc task. The system is based on the vector space model. Formally, a document is represented as a list of terms or term vectors. A document collection is represented as a term-document matrix which is normally very sparse. A query consists of a list of terms, too.

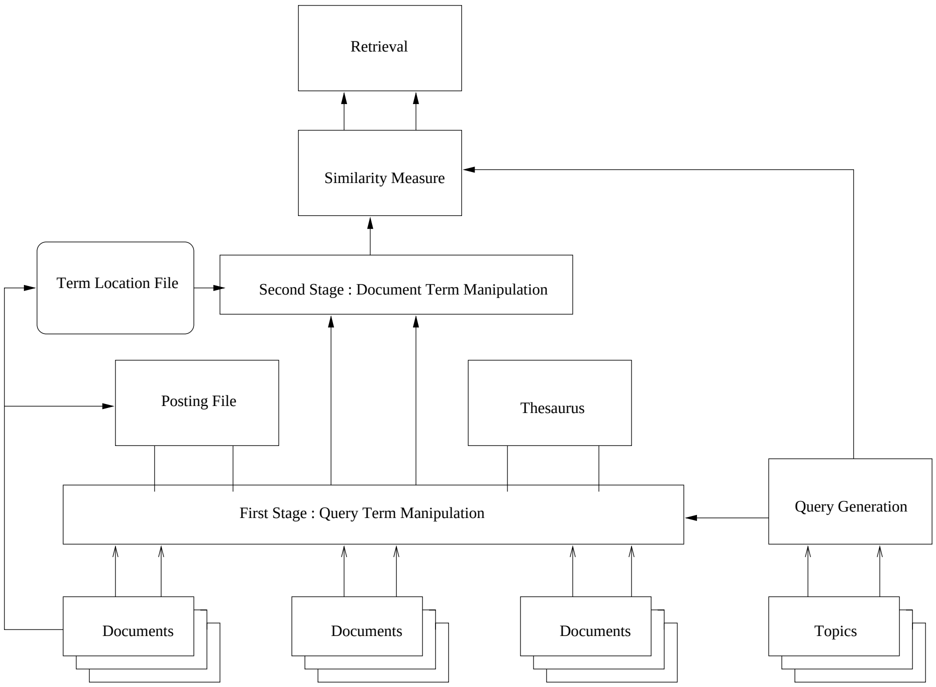

Figure 1: The two-stage model for text retrieval.

The documents are indexed by using the classical $tf \cdot idf$ weighting scheme [7]:

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

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.

We have used various stemming methods for TREC-7 data, but could not achieve any significant performance improvement. For this reason, we used all words as indexing terms that appeared in raw documents but not in the stop-list, and then tried to improve recall and precision in sequence. In the first stage, operations are carried out with respect to query terms, while in the second stage operations are performed with respect to document terms. The main objective of the first stage is to improve recall, while the second stage aims mainly at improving precision.

Techniques used in the first stage include query expansion methods based on a modified stemming algorithm and the WordNet. Techniques adopted in the second stage include methods using a term proximity-based scoring function and a prototype-based ranking method.

3 Retrieval Methods

3.1 Query Expansion

Two methods for query expansion have been studied. One is substring matching in query generation. The original query terms are generated from the topic text in the TREC-7 documents using the same method as for document indexing. Query j for topic j is then represented as a vector $\mathbf{v}_j = (v_{j1}, ..., v_{jk}, ..., v_{jn})$, where v_{jk} denotes the weight of term k in query j. In the substring matching method, the weight v_{jk} of term k is assigned proportional to the document frequency df_k of the corresponding term. The idea behind this weighting scheme is that people tend to use low-frequency terms in queries, and thus document frequency information is important. Note that this is a $tf \cdot idf$ method modified for query indexing.

We also studied a query expansion method using a thesaurus. The original query terms v_{jk} are expanded by their synonyms and hypernyms which are found using the WordNet [4].

3.2 Query Weighting

Once the queries are generated, they are matched against documents as follows. Let $\mathbf{w}_i = (w_{ik})$ and $\mathbf{v}_j = (v_{jk})$ denote the term vectors for document i and query j, where w_{ik} and v_{jk} are weight values for kth term in document and query, respectively. The relevance of document i with respect to query j is scored by the inner product of the document and query vectors:

$$S_j(i) = \mathbf{w}_i \cdot \mathbf{v}_j = \sum_{k=1}^{n} w_{ik} \cdot v_{jk}, \tag{2}$$

where k runs over the terms in the vocabulary of size n.

In a modified version, we use query weighting after query expansion. Here we regard the terms with low document frequency more important. To implement this, query terms are ranked and then the importance of term k is weighted by a power function

$$(m - rank_{jk})^p, \tag{3}$$

where m is the number of terms in query j and $rank_{jk}$ is the rank of the kth term in query j. Large p gives more weight to the ranking factor. In effect, the similarity score of document i to query j is defined as:

$$S_j(i) = \sum_{k=1}^{n} w_{ik} \cdot v_{jk} \cdot (m - rank_{jk})^p. \tag{4}$$

The lower the frequency of query terms is, the greater the score of the document. The score can be adjusted by the p value.

3.3 Term Proximity Information

The second stage aims at improving the retrieval precision. We experimented with two methods. One is using the word proximity information. Though appearing in the same document, two different terms may have no relationship with each other if one term occur far away from the other. To use proximity information, we apply an additional query operator, called NEAR, that take into account the distance between terms in the document. The proximity $prox(r, s)$ of rth and sth terms in document i is then defined as

$$prox(r, s) \propto \frac{1}{dist(w_{ir}, w_{is})} \tag{5}$$

where $dist(\cdot)$ is a distance measure. The proximity score $prox(i)$ of ith document is then defined as the sum of the proximity of term pairs in the document:

$$prox(i) = \sum_r \sum_s prox(r, s) \tag{6}$$

where r and s run over the terms in the ith document.

3.4 Prototype-Based Reranking

The second method for improving precision is to use the documents retrieved to rerank them. Among the retrieved documents, we select the top K documents which are then used to construct prototype documents. Let $\mathbf{p}_j = (p_{j1}, ..., p_{jn})$ denote the weight vector of the jth prototype, where n is the number of indexing terms. The similarity of document i to prototype j is then measured by cosine coefficient:

$$sim(i, j) = \frac{\sum_{k=1}^{n} w_{ik} \cdot p_{jk}}{\sqrt{\sum_{k=1}^{n} {w_{ik}}^2 \cdot \sum_{k=1}^{n} {p_{jk}}^2}} \tag{7}$$

where w_{ik} and p_{jk} are term weights for document i and prototype j, respectively.

4 Experimental Results

The methods described in the previous section have been used in various combinations for the ad hoc query on TREC-7 collections.

Table 1 summarizes the experimental results. The first row, i.e. experiment number 1, shows the results of the baseline retrieval method. This is the results we submitted to NIST in the summer of 1998. After this official submission, we extended the system by the techniques described in the previous section. Figure 2 shows the recall-precision curves for the methods tested.

Rows 2 and 3 in the table show the results obtained by using the query weighting method. In experiment 2, long queries were used, i.e. queries were generated from the title, description, and narrative fields of the topic text. Experiment 3 used short queries, i.e. the title field only. Term

Experiment No.	Run Type	Topic Length	Average Precision
1	baseline	T+D+N	0.0477
2	qwgt1	T+D+N	0.0601
3	qwgt2	T	0.0903
4	qexp1 (WordNet)	T	0.0967
5	qexp2 (range 1)	T	0.0778
6	qexp3 (range 3)	T	0.0640
7	proto	T	0.0652
8	proto + prox	T	0.1258
9	qwgt + prox	T	0.1468
10	qwgt + prox + proto	T	0.1277

Table 1: Comparison of average precisions for various combinations of methods. Symbols denote the names of various techniques: qwgt = query weighting, qexp = query expansion, prox = proximity infomation, proto = prototype-based ranking.

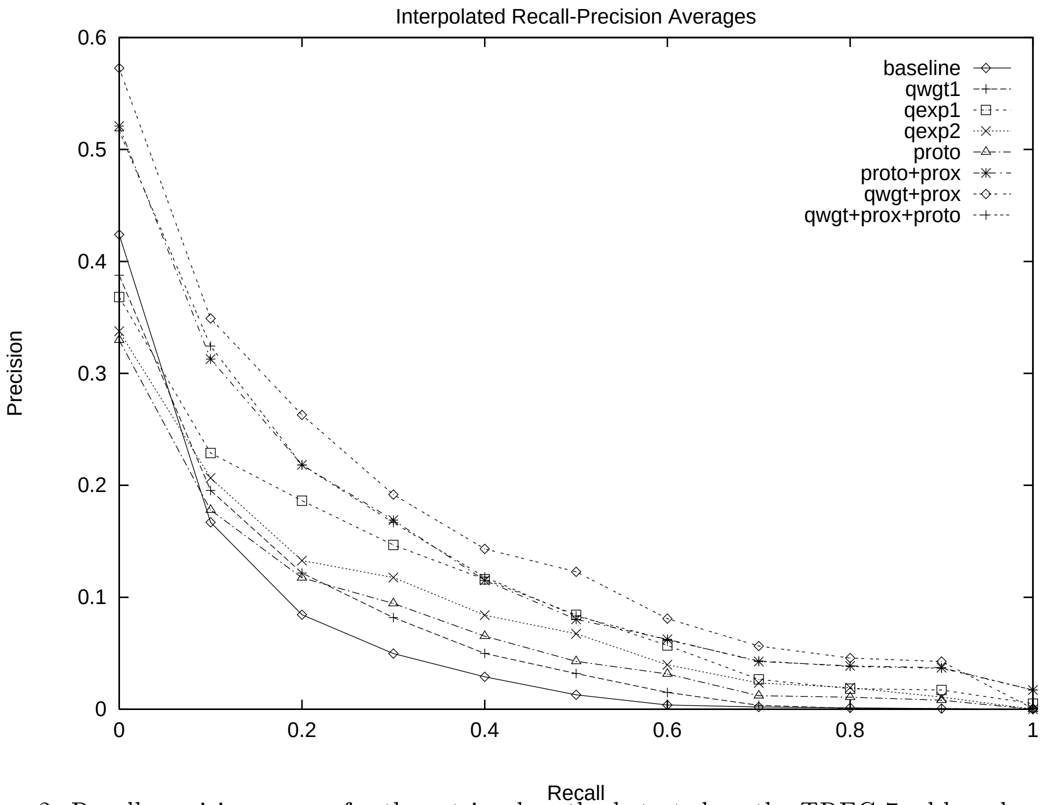

Figure 2: Recall-precision curves for the retrieval methods tested on the TREC-7 ad hoc document collection.

weighting improved the baseline method and, in terms of query length, short queries were better than long ones.

Rows 4 to 6 in the table summarize the performances of query expansion. Compared are three variants: using WordNet (experiment 4), substring matching with range of 1 (experiment 5), substring matching with range of 3 (experiment 6). Among these, the WordNet-based expansion method was the best. In substring matching, increasing the range of matching degraded the precision.

Experiments 7 to 10 are related with using proximity information and prototype-based reranking. In general, these methods and their combinations improved the precision. In particular, the effect of proximity information was more significant than the others. This is due to the fact that the term proximity measure extracts context information of terms, which is used as an additive term to the scoring function. In contrast, the reranking based on prototypes did not lead to significant improvements. This seems attributed to the fact that we used in these experiments one prototype constructed as the average of K document vectors rather than multiple prototypes; It is not very likely that the average pattern of top K (in our case 20) documents is representative of all the documents in the topic class.

5 Conclusions

We presented a two-stage model for the TREC-7 ad hoc retrieval task. By dividing the retrieval process into two stages, we could reduce the complicated interference effect of recall and precision on the whole performance. We proposed and experimented with various techniques that were designed to improve recall and precision, respectively.

Tested on the TREC-7 ad hoc text data, we improved the average precision performance from 0.0477 for the baseline method to 0.1468 for the two-stage method combining query manipulation and term proximity information. Though this performance is not among the best of the TREC-7 ad hoc entries in absolute value, we think it is a significant improvement as our first experiments in TREC. Refinements of proposed methods are in progress to further improve the performance of the current retrieval system.

Acknowledgements

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

References

[1] Frakes, W.B and Ricardo, Baeza-Yates, *Information Retrieval*, Prentice-Hall, 1992.

[2] Korfhage, Robert R., *Information Storage And Retrieval*, John Wiley & Sons, 1997.

[3] Lee, J.H., Analyses of Multiple Evidence Combination, *SIGIR-97*, pp. 267-276, 1997.

[4] Miller, G.A., Five papers on WordNet, *International Journal of Lexicology*, vol. 3, no. 4, 1990.

[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] Salton, G., *Automatic Text Processing*, Addison-Wesley, 1989.

[8] Yang, Y., Noise Reduction in a Statistical Approach to Text Categorization, *SIGIR-95*, pp. 256-263, 1995.