Combining Multiple Evidence from Different Relevance Feedback Methods

Joon Ho Lee*
Center for Intelligent Information Retrieval
Department of Computer Science, University of Massachusetts
Amherst, Massachusetts 01003, USA

Abstract

It has been known that using different representations of a query retrieves different sets of documents. Recent work suggests that significant improvement in retrieval performance can be achieved by combining multiple representations of an information need. In this paper, we first investigate a fully automatic way of generating multiple query representations for a given information problem. We produce multiple query vectors by expanding an initial query vector with various relevance feedback methods. We then describe the effect of combining the multiple query vectors on retrieval effectiveness. Experimental results show that significant improvements can be obtained by the combination of multiple query vectors expanded with different relevance feedback methods.

1 Introduction

A variety of representation techniques for queries and documents has been proposed in the information retrieval literature, and many corresponding retrieval techniques have also been developed to get higher effectiveness of information retrieval. Recent research shows that retrieval effectiveness can be improved by using multiple query or document representations, or multiple retrieval techniques, and combining the retrieval results, in contrast to using just a single representation or a single retrieval technique. This general area has been discussed in the literature under the name of "data fusion".

McGill, Koll & Norreault [10] found that there was surprisingly little overlap between document sets for the same information need, when documents were retrieved by different users or by the same user using controlled versus free-text vocabularies. Katzer, et al. [7] considered the effect of different document

*On leave from Korea Research and Development Information Center, Korea Institute of Science and Technology, P.O. Box 122, Yusong, Taejon 305-600, Korea

representations (e.g. title, abstract) on retrieval effectiveness rather than different query representations. They discovered the same phenomenon that the various document representations gave similar retrieval effectiveness, but retrieved quite different sets of documents. These results suggest that the combined run may retrieve more relevant documents than any individual run, therefore providing higher recall.

Saracevic and Kantor [17] asked different experts to construct Boolean queries based on the same description of information problem in operational online information retrieval systems. Once again, they found that different query formulations generated different documents. They, however, noticed that the odds of a document being judged relevant increase monotonically with the number of retrieved sets in which the document appears. If the combining method is designed to favor the documents retrieved by more retrieval runs, the combined run can result in more accurate similarity values between queries and documents, and therefore give higher precision.

Turtle and Croft [18] developed an inference network-based retrieval model, which can combine different document representations and different versions of a query in a consistent probabilistic framework. Turtle and Croft implemented the INQUERY retrieval system based on the model, and demonstrated that multiple evidence increases retrieval effectiveness in some circumstances. Fox and Shaw [5] have worked on various methods for combining multiple retrieval runs, and have obtained improvements over any single retrieval run. Belkin, et al. [1] showed that progressive combination of different Boolean query formulations could lead to progressive improvements of retrieval effectiveness. Lee [9] described how different properties of weighting schemes may retrieve different types of documents, and showed that significant improvements could be obtained by combining the retrieval results from different properties of weighting schemes.

The research results described above show that combining multiple retrieval runs can improve the effectiveness of information retrieval. The aim of our study is to propose a fully automatic way of generating multiple query representations for a given information problem, and to investigate the effect of combining the multiple query representations on retrieval effectiveness. We first generate an initial query vector for a given information problem, and perform the initial retrieval. Second, the top-retrieved documents are all assumed to be relevant, and multiple query vectors are generated by applying various relevance feedback methods. Then, we perform the feedback runs, and combine the retrieval results.

The basic idea is that different relevance feedback methods might have different properties, and might generate quite different new query vectors. In order to confirm this idea, we calculate the similarity between the query vectors expanded with different relevance feedback methods. We also analyze through the Spearman correlation coefficient and the number of common documents that the expanded query vectors retrieve different documents. Experiments on combining the retrieval results suggest that improvements could be achieved by the combination of multiple query vectors expanded with different relevance feedback methods.

The remainder of this paper is organized as follows. Section 2 gives the description of the SMART system that is used to perform our experiments. Section 3 describe various relevance feedback methods. In section

4, we propose a fully automatic way of generating multiple query vectors for a given information problem, and analyze various aspects of the multiple query vectors. In section 5, we combine the results retrieved by the multiple query vectors. Finally, concluding remarks are given in section 6.

2 The SMART System

The SMART system [13] has been developed at Harvard and Cornell Universities for over 35 years. The indexing of both queries and documents is completely automatic, and therefore human experts are not required for either the initial collection creation or the actual query formulation. This means that retrieval results are reasonably collection independent and should be valid across a wide range of collections.

SMART is based on the vector space model [15], and transforms the description of information problems as well as the stored documents into vectors of the form:

$$d_i = (w_{i1}, w_{i2}, \ldots, w_{in})$$

where d_i represents a document (or query) text and w_{ik} is the weight of term t_k in document d_i. The assumption is that n terms in all are available for the representation of queries and documents. A weight of zero is used for terms that are absent from a particular document, and positive weights characterize terms actually assigned for content identification. In the SMART context, such vectors are formed by a text transformation as follows:

1. recognize individual text words
2. eliminate function words with a stop list
3. generate word stems by removing suffixes
4. assign term weights to all remaining word stems to form the term vector

Even though SMART can handle term phrases formed with statistical word co-occurrence or syntactic analysis, we will be concerned only with word stems in generating query and document vectors in this paper.

Once term vectors are available for all information items, all subsequent processing is based on term vector manipulations. When document d is represented by a vector of the form (w_{d1}, w_{d2}, ..., w_{dn}) and query q by the vector (w_{q1}, w_{q2}, ..., w_{qn}), the similarity between document d and query q is calculated as the inner product between corresponding weighted term vectors as follows:

$$Sim(d, q) = \sum_{i=1}^{n} (w_{di} \times w_{qi})$$

The query-document similarity depends on the weights of coinciding terms in the two vectors, and therefore the term weighing scheme is an important factor affecting the effectiveness of SMART.

In constructing a term weighting scheme, three main components such as term frequency, collection frequency and normalization have been considered in the information retrieval literature [14]. First, the term frequency component assigns higher weights to the terms that occur more frequently in the text. Second, the collection frequency component assigns higher weights to the terms that occur in fewer documents of the collection. The normalization component equalizes the length of document vectors in the collections with varying document vector length.

In the SMART system, a term weighting scheme is described by using two triples representing the term frequency, collection frequency and normalization. The first and second triples are for document terms and query terms, respectively. In this paper, we use the $lnc \cdot ltc$ system, which gives high retrieval effectiveness for the TREC data collections. The ltc weighting scheme for query term weights uses cosine normalization of logarithmic term frequency $\times$ inverse document frequency as follows:

$$w_{ik} = \frac{(\log(tf_{ik}) + 1.0) * \log(N/n_k)}{\sqrt{\sum_{j=1}^{n} [(\log(tf_{ij}) + 1.0) * \log(N/n_j)]^2}}$$

where tf_{ik} is the occurrence frequency of term t_k in query q_i, N is the total number of documents in the collection, and n_k is the number of documents to which term t_k is assigned. The documents are weighted with the lnc weighting scheme which is the same as the ltc query scheme, except no inverse document frequency factor is used.

3 Relevance Feedback Methods

The relevance feedback process is an automatic process for query reformulation [16]. The main idea consists in choosing important terms in relevant documents, and of enhancing the weight of these terms in a new query formulation. Analogously, terms included in previously retrieved nonrelevant documents could be deemphasized in any future query formulation. The effect of such a query modification process is to "move" the query in the direction of the relevant items and away from the nonrelevant ones, in the expectation of retrieving more wanted and fewer nonwanted items in a later search.

The original relevance feedback process was designed to be used with vector queries, that is, query statements consisting of sets of possibly weighted search terms used without Boolean operators. A particular search expression might then be written as

$$q = (w_{q1}, w_{q2}, \ldots, w_{qn})$$

where w_{qi} represents the weight of term t_i in query q. The term weights are often restricted to the range from 0 to 1, where 0 represents a term that is absent from the vector, and 1 represents a fully weighted term.

Given a query vector of the type shown above, the relevance feedback process generates a new query

vector

$$q' = (w'_{q1}, w'_{q2}, \ldots, w'_{qn})$$

where w'_{qi} represents altered term weights for index term t_i. New terms are introduced by assigning a positive weight to terms with an initial weight of 0, and old terms are deleted by reducing to 0 the weight of terms that were initially positive.

In this paper, we use relevance feedback methods to generate multiple query vectors for a given information need. Although a variety of relevance feedback methods have been developed in the information retrieval literature, five different relevance feedback methods implemented in SMART version 11.10 are exploited for evaluation purposes, including two vector modification methods and three probabilistic feedback methods. They are described in the following.

Rocchio The new query vector Q_{new} is the vector sum of the old query vector plus the vectors of the relevant and nonrelevant documents [12].

$$Q_{new} = \alpha \cdot Q_{old} + \beta \cdot \sum_{r=1}^{n_{rel}} \frac{D_r}{n_{rel}} - \gamma \cdot \sum_{n=1}^{n_{nonrel}} \frac{D_n}{n_{nonrel}}$$

α, β, γ constants

D_r vector for rel doc d_r

D_n vector for nonrel doc d_n

n_{rel} number of rel docs

n_{nonrel} number of nonrel docs

Ide The Ide formula is modified from the Rocchio formula by eliminating the normalization for the number of relevant and nonrelevant documents and allowing limited negative feedback from only the top-ranked nonrelevant document [4].

$$Q_{new} = \alpha \cdot Q_{old} + \beta \cdot \sum_{i=1}^{n_{rel}} R_i - \gamma \cdot T_{nonrel}$$

T_{nonrel} vector for a top-ranked nonrel doc

Pr_cl This classical probabilistic feedback formula is based on the probabilistic retrieval model [3].

$$w'_{qi} = \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

$$p_i = \frac{r_i + 0.5}{R+1}$$

$$q_i = \frac{n_i - r_i + 0.5}{N - R + 1}$$

r_i number of rel docs having term t_i

n_i number of docs having term t_i in the collection

R total number of rel docs

N number of docs in the collection

Pr_adj This adjusted probabilistic feedback formula is modified from the Pr_cl formula by replacing the 0.5 adjustment factor with n_i/N [11].

$$w'_{qi} = \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

$$p_i = \frac{r_i + n_i/N}{R+1}$$

$$q_i = \frac{n_i - r_i + n_i/N}{N-R+1}$$

S_rpi This is a simplified version of Fuhr's RPI formula that does not use a non-linear similarity function [6].

$$w'_{qi} = \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

$$p_i = \sum_{r=1}^{n_{rel}} \frac{w_{ri}}{n_{rel}}$$

$$q_i = \sum_{n=1}^{n_{nonrel}} \frac{w_{ni}}{n_{nonrel}}$$

w_{ri} weight of term t_i in rel doc d_r

w_{ni} weight of term t_i in nonrel doc d_n

n_{rel} number of rel docs

n_{nonrel} number of nonrel docs

4 Generating Multiple Evidence

In this section we describe a completely automatic way of generating multiple query representations for a given information problem. The basic idea is that different relevance feedback methods might have different properties, and might generate quite different new query vectors. In order to confirm this basic idea, we have performed the experiments known as query expansion without relevance information as follows:

1. generate an initial query vector for a given information problem.
2. perform the initial retrieval, and assume the top-retrieved 30 documents are relevant.
3. generate new multiple query vectors by applying various relevance feedback methods.

4. normalize the new query vectors with cosine normalization where each term weight is divided by a factor representing Euclidian vector length.

5. perform the feedback retrievals with the new query vectors.

We have exploited the five relevance feedback methods described in the previous section. We evaluated the various runs with the TREC collections, namely TREC D1& D2. We retrieve the top-ranked 1000 documents for 50 queries, namely TREC topics 151-200, and evaluate the performance using 11-point average precision. The results presented in Table 1 show that the feedback runs provide much better effectiveness than the initial run.

We calculated query-query similarities as the inner product between corresponding query vectors. Since cosine normalization is used to normalize the query vectors, the similarity takes any value between zero and one, i.e. $0 \leq Sim(q_i, q_j) \leq 1$. The similarity is 1 for identical query vectors. Table 2 shows the similarities between the initial query vector and the expanded query vectors, and Table 3 shows the similarities between the expanded query vectors. We can easily see that different relevance feedback methods generate quite different new query vectors even though the new query vectors result in similar level of retrieval effectiveness.

It is well known that different query representations could retrieve different sets of documents. We proposed a method for generating multiple query vectors for a given information problem. In what follows, we analyze the results retrieved by the multiple query vectors so that we show the multiple query vectors retrieve different documents. Two different methods are exploited to calculate how different the retrieval results are. First of all, we compute the Spearman correlation coefficient to see how two ranked lists are correlated. The Spearman correlation coefficient ρ [8] is defined as follows:

> Suppose k documents such as $d_1, \ldots, d_k$ are given. The Spearman correlation coefficient ρ between two ranked output $r_1, \ldots, r_k$ and $r'_1, \ldots, r'_k$ is given by
>
> $$\rho = 1 - 6 \times \left(\frac{\sum_{i=1}^{k} (r'_i - r_i)^2}{k(k^2 - 1)} \right)$$
>
> The coefficient is 1 for identical ranked output, 0 for unrelated ranked output, and -1 for inversely related ranked output.

TREC D1 & D2 contain more than 740,000 documents. It would require much effort to fully rank all the documents for the expanded query vectors and to calculate the Spearman correlation coefficient. Hence, we ranked only relevant documents, and applied the rank correlation method to pairwise combinations of the five feedback runs. Table 4 shows the Spearman correlation coefficients between the five feedback runs.

The more different expanded query vectors are, the more different documents they should retrieve. That is, the greater the similarity value between expanded query vectors is, the greater the correlation coefficient

between the retrieval results should be. The number in parentheses is the rank in decreasing order of query-query similarities and correlation coefficients in Table 3 and Table 4, respectively. We should note that the ranking in Table 3 is highly related to that in Table 4.

In ranked output systems, the systems usually retrieve some top-ranked documents. We propose a simpler method than the rank correlation method, which can estimate how two ranked output is correlated. We count the number of common documents retrieved by two expanded query vectors, which is shown in Table 5. The number in parentheses is the rank in decreasing order of the numbers of common documents. We can easily see that the number of common documents is a good predictor of the correlation coefficients by comparing the ranking in Table 4 with that in Table 5.

5 Combining Multiple Evidence

It has been known that if two runs retrieve different sets of documents, significant improvements can be achieved by combining the retrieval results. In the previous section, we have shown that the different query vectors expanded with different feedback methods retrieve quite different documents even though the different runs provide similar level of retrieval effectiveness. In this section, we combine the retrieval results generated by the expanded query vectors. First of all, we give the description of the combining method, and then present the combined results.

Since different retrieval runs generate quite different ranges of similarity values, a normalization method should be applied to each retrieval result. We normalize each similarity value by the maximum similarity value in a retrieval result, which will be called Max_Norm.

$$Max\_Norm = \frac{old\_sim}{maximum\_sim}$$

Basically, normalization plays the role of controlling the ranges of similarities that retrieval systems generate. Max_Norm aligns only the upper bound of similarities. Hence, in order to even the lower bound as well as the upper, the following Min_Max_Norm looks more reasonable than Max_Norm.

$$Min\_Max\_Norm = \frac{old\_sim - minimum\_sim}{maximum\_sim - minimum\_sim}$$

The minimum similarity generated by SMART is zero, in that SMART gives zero to the documents that do not have terms specified in a query. Therefore, Min_Max_Norm can be reduced to Max_Norm for SMART.

Fox and Shaw [5] have tested several functions of combining similarity values with the SMART system. As a result, the summation function, which sums up the set of similarity values, or, equivalently, the numerical mean of the set of similarity values works better in most TREC subcollections. Belkin, et al. [1] used the summation function to combine multiple Boolean query representations of TREC topics, which is supported by the INQUERY system. In this paper we also use the summation function for the combination of retrieval

Table 1: 11-point average precision of query expansion without relevance information (TREC D1 & D2; averages over 50 queries)

Initial	Ide	Rocchio	Pr_cl	Pr_adj	S_rpi
0.2893	0.3523 (+21.8%)	0.3482 (+20.4%)	0.3361 (+16.2%)	0.3378 (+16.8%)	0.3301 (+14.1%)

Table 2: The similarity between the initial query vector and the expanded query vectors (TREC D1 & D2; averages over 50 queries)

	Ide	Rocchio	Pr_cl	Pr_adj	S_rpi
Initial	0.5803	0.9566	0.2742	0.2522	0.2387

Table 3: The similarity between the expanded query vectors (TREC D1 D2; averages over 50 queries)

	Ide	Rocchio	Pr_cl	Pr_adj
Rocchio	0.7900 (4)			
Pr_cl	0.6746 (5)	0.4456 (8)		
Pr_adj	0.6595 (6)	0.4239 (9)	0.9856 (1)	
S_rpi	0.6470 (7)	0.4091 (10)	0.9725 (2)	0.9403 (3)

Table 4: The Spearman correlation coefficient between the ranked output retrieved by the expanded query vectors (TREC D1 & D2; the ranked output is generated for only relevant documents)

	Ide	Rocchio	Pr_cl	Pr_adj
Rocchio	0.9355 (4)			
Pr_cl	0.8948 (6)	0.8029 (9)		
Pr_adj	0.9019 (5)	0.8114 (8)	0.9974 (1)	
S_rpi	0.8890 (7)	0.7951 (10)	0.9898 (2)	0.9891 (3)

Table 5: Number of common documents retrieved by the expanded query vectors (TREC D1 & D2; top-ranked 1000 documents are retrieved for 50 queries)

	Ide	Rocchio	Pr_cl	Pr_adj
Rocchio	737.58 (4)			
Pr_cl	655.32 (6)	530.64 (9)		
Pr_adj	668.54 (5)	539.54 (8)	985.50 (1)	
S_rpi	642.36 (7)	516.72 (10)	881.06 (2)	880.82 (3)

results as follows:

$$combined\_sim = SUM(individual\_sims)$$

We applied the combining method to pairwise combinations of the five feedback runs using different relevance feedback methods. Performance results of the combined runs are presented in Table 6, in which % change is given with respect to the initial run providing the 11-point average precision 0.2893. The table shows that the combined runs provide better retrieval effectiveness that the feedback runs. These results show that significant improvements might be achieved by combining multiple query vectors expanded with different relevance feedback methods.

The more different documents two different runs retrieve, the more improvements their combination should result in. In order to confirm this natural conjecture, we need to calculate how much improvements the combined run gives over the individual runs participating in the combination. We computed % changes with respect to the run providing better effectiveness in the combination, which is presented in Table 7. The results in Table 7 seem to have little coincidence with those in Table 4, in that they do not seem to agree with the conjecture. For example, when we combine Ide and Pr_adj, the correlation coefficient is 0.9019 and the improvement is +2.3%. However, at the combination of Ide and S_rpi, we get +1.6% even though the correlation coefficient 0.8890 is a bit less than 0.9019.

It should be noticed that the effectiveness of combined runs is affected by the effectiveness of individual runs as well as their correlation. In combining Ide and S_rpi, smaller improvement may be due to the fact that the effectiveness of S_rpi, i.e. 0.3301 is lower than that of Pr_adj, i.e. 0.3378. Hence, we computed % changes with respect to the average of the effectiveness that the individual runs provide, which is presented in Table 8. We can see that the results in Table 8 have more coincidence with those in Table 4, in that the less correlation coefficient two different runs have, the more improvements their combination gives.

Our final investigation in the combination of multiple query vectors is to see if combining more than two feedback runs has a beneficial effect. We combined not only all pairwise combinations of feedback runs (called 2-way from now on) but also all 3-way combinations of feedback runs, all 4-way combinations of feedback runs, and the combination of all 5 feedback runs. Table 9 shows the 11-point average precision for the average of all combined runs in each level of combination and for the best performing combination at each level. The table shows that the average and worst performance increases monotonically as more evidence is added. However, when one takes the best performing combination, 2-way and 3-way combination is better than 1-way, 4-way, and 5-way. These are the same results that Belkin, et al. obtained in the combination of different Boolean query formulations [2].

Table 6: 11-point average precision of the combined runs (TREC D1 & D2; averages over 50 queries; % change is given with respect to the initial run providing the 11-point average precision 0.2893)

	Ide 0.3523 (+21.8%)	Rocchio 0.3482 (+20.4%)	Pr_cl 0.3361 (+16.2%)	Pr_adj 0.3378 (+16.8%)
Rocchio 0.3482 (+20.4%)	0.3529 (+22.0%)			
Pr_cl 0.3361 (+16.2%)	0.3602 (+24.5%)	0.3659 (+26.5%)		
Pr_adj 0.3378 (+16.8%)	0.3604 (+24.6%)	0.3666 (+26.7%)	0.3376 (+16.7%)	
S_rpi 0.3301 (+14.1%)	0.3578 (+23.7%)	0.3655 (+26.3%)	0.3335 (+15.3%)	0.3342 (+15.5%)

6 Conclusion

Various strategies for representing queries and documents, and various retrieval techniques are available these days in the information retrieval literature. Several researchers have investigated the effect of combining multiple representations of either queries or documents, or multiple retrieval techniques on retrieval performance because different representations or different retrieval techniques can retrieve different documents. Recent work has shown that significant improvements can be achieved by the combination of multiple evidence.

In this paper we have proposed a completely automatic method for generating multiple query vectors for a given information problem. The method can be easily incorporated in the system using a single query representation, a single document representation and a single retrieval technique. We first generated an initial query vector, and performed the initial retrieval. Second, the top-retrieved documents were all assumed to be relevant, and generated multiple query vectors by applying various relevance feedback methods. Then, we performed the feedback runs, and combined the retrieval results. Experimental results suggest that improvements could be achieved by the combination of multiple query vectors expanded with different relevance feedback methods.

We have generated multiple query vectors for a given information problem in an ad-hoc situation. We can also generate multiple query vectors in routing environment, and combine the retrieval results in order to get higher retrieval effectiveness in the same manner.

Table 7: 11-point average precision of the combined runs (TREC D1 & D2; averages over 50 queries; % change is given with respect to the run providing better effectiveness in the combination)

	Ide 0.3523	Rocchio 0.3482	Pr_cl 0.3361	Pr_adj 0.3378
Rocchio 0.3482	0.3529 (+0.1%)			
Pr_cl 0.3361	0.3602 (+2.2%)	0.3659 (+5.1%)		
Pr_adj 0.3378	0.3604 (+2.3%)	0.3666 (+5.3%)	0.3376 (-0.1%)	
S_rpi 0.3301	0.3578 (+1.6%)	0.3655 (+5.0%)	0.3335 (-0.7%)	0.3342 (-1.1%)

Table 8: 11-point average precision of the combined runs (TREC D1 & D2; averages over 50 queries; % change is given with respect to the average of the individual runs participating in the combination)

	Ide 0.3523	Rocchio 0.3482	Pr_cl 0.3361	Pr_adj 0.3378
Rocchio 0.3482	0.3529 (+0.8%)			
Pr_cl 0.3361	0.3602 (+4.6%)	0.3659 (+6.9%)		
Pr_adj 0.3378	0.3604 (+4.4%)	0.3666 (+6.9%)	0.3376 (+0.2%)	
S_rpi 0.3301	0.3578 (+4.9%)	0.3655 (+7.8%)	0.3335 (+0.1%)	0.3342 (+0.1%)

Table 9: 11-point average precision for the average of all combined runs in each level of combination (average) and for the best performing combination at each level (best) (TREC D1 & D2; averages over 50 queries)

	Initial	1-way	2-way	3-way	4-way	5-way
average	0.2893	0.3409 (+17.8%)	0.3535 (+22.2%)	0.3565 (+23.2%)	0.3587 (+24.0%)	0.3582 (+23.8%)
best	0.2893	0.3523 (+21.8%)	0.3666 (+26.7%)	0.3684 (+27.3%)	0.3653 (+26.3%)	0.3582 (+23.8%)

Acknowledgments

This work was supported in part by the NSF Center for Intelligent Information Retrieval at the University of Massachusetts at Amherst. I would like to thank James Allan and Jamie Callan for reading this paper. Their comments gave me a future direction of this work. I would also like to give special thanks to Gerard Salton and Chris Buckley who have developed the SMART system.

References

[1] N.J. Belkin, C. Cool, W.B. Croft and J.P. Callan, "The effect of multiple query representations on information retrieval performance," Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 339-346, 1993.

[2] N.J. Belkin, P. Kantor, E.A. Fox and J.A. Shaw, "Combining the evidence of multiple query representations for information retrieval," Information Processing & Management, Vol. 31, No. 3, pp. 431-448, 1995.

[3] W.B. Croft and D.J. Harper, "Using probabilistic models of document retrieval without relevance," Journal of Documentation, Vol. 35, pp. 285-295, 1979.

[4] E. Ide, "New experiments in relevance feedback," The Smart system – experiments in automatic document processing, Englewood Cliffs, NJ: Prentice Hall Inc., pp. 337-354, 1971.

[5] E.A. Fox and J.A. Shaw, "Combination of multiple searches," Proceedings of the 2nd Text REtrieval Conference (TREC-2), National Institute of Standards and Technology Special Publication 500-215, pp. 243-252, 1994.

[6] N. Fuhr and C. Buckley, "A probabilistic learning approach for document indexing," ACM Transactions on Information Systems, Vol. 9, No. 3, pp. 223-248, 1991.

[7] J. Katzer, M.J. McGill, J.A. Tessier, W. Frakes and P. Dasgupta, "A study of the overlap among document representations," Information Technology: Research and Development, Vol. 1, No. 2, pp. 261-274, 1982.

[8] J.H. Lee, M.H. Kim and Y.J. Lee, "Ranking documents in thesaurus-based Boolean retrieval systems," Information Processing & Management, Vol. 30, No. 1, pp. 79-91, 1994.

[9] J.H. Lee, "Combining Multiple Evidence from Different Properties of Weighting Schemes," Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 180-188, 1995.

[10] M. McGill, M. Koll and T. Norreault, "An evaluation of factors affecting document ranking by information retrieval systems," Syracuse, Syracuse University School of Information Studies, 1979.

[11] S.E. Robertson, "On relevance weight estimation and query expansion," Journal of Documentation, Vol. 42, pp. 182-188, 1986.

[12] J.J.Jr. Rocchio, "Relevance feedback in information retrieval," The Smart system – experiments in automatic document processing, Englewood Cliffs, NJ: Prentice Hall Inc., pp. 313-323, 1971.

[13] G. Salton and M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., 1983.

[14] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," Information Processing and Management, Vol. 24, No. 5, pp. 513-523, 1988.

[15] G. Salton, Automatic Text Processing - The Transformation, Analysis and Retrieval of Information by Computer, Addison-Wesley Publishing Co., Reading, MA, 1989.

[16] G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," Journal of the American Society for Information Science, Vol. 41, No. 4, pp. 288-297, 1990.

[17] T. Saracevic and P. Kantor, "A study of information seeking and retrieving. III. Searchers, searches, overlap," Journal of the American Society for Information Science, Vol. 39, No. 3, pp. 197-216, 1988.

[18] H. Turtle and W.B. Croft, "Evaluation of an inference network-based retrieval model," ACM Transactions on Information Systems, Vol. 9, No. 3, pp. 187-222, 1991.