Query Expansion with the Minimum Relevance Judgments

Masayuki Okabe[1], Kyoji Umemura[2], and Seiji Yamada[3]

[1] Information and Media Center, Toyohashi University of Technology,
Tempaku 1-1, Toyohashi, Aichi, Japan
okabe@imc.tut.ac.jp
[2] Information and Computer Science, Toyohashi University of Technology,
Tempaku 1-1, Toyohashi, Aichi, Japan
umemura@tutics.tut.ac.jp
[3] National Institute for Informatics, Chiyoda, Tokyo, Japan
seiji@nii.ac.jp

Abstract. Query expansion techniques generally select new query terms from a set of top ranked documents. Although a user's manual judgment of those documents would much help to select good expansion terms, it is difficult to get enough feedback from users in practical situations. In this paper we propose a query expansion technique which performs well even if a user notifies just a relevant document and a non-relevant document. In order to tackle this specific condition, we introduce two refinements to a well-known query expansion technique. One is to increase documents possibly being relevant by a transductive learning method because the more relevant documents will produce the better performance. The other is a modified term scoring scheme based on the results of the learning method and a simple function. Experimental results show that our technique outperforms some traditional methods in standard precision and recall criteria.

1 Introduction

Query expansion is a simple but very useful technique to improve search performance by adding some terms to an initial query. While many query expansion techniques have been proposed so far, a standard method of performing is to use relevance information from a user [1]. If we can use more relevant documents in query expansion, the likelihood of selecting query terms achieving high search improvement increases. However it is impractical to expect enough relevance information. Some researchers said that a user usually notifies few relevance feedback or nothing [2].

In this paper we investigate the potential performance of query expansion under the condition that we can utilize little relevance information, especially we only know a relevant document and a non-relevant document. To overcome the lack of relevance information, we tentatively increase the number of relevant documents by a machine learning technique called *Transductive Learning*. Compared with ordinal inductive learning approach, this learning technique works

even if there is few training examples. In our case, we can use many documents in a hit-list, however we know the relevancy of few documents. When applying query expansion, we use those increased documents as if they were true relevant ones.

The point of our query expansion method is that we focus on the availability of relevance information in practical situations. There are several researches which deal with this problem. Pseudo relevance feedback which assumes top n documents as relevant ones is one example. This method is simple and relatively effective if a search engine returns a hit-list which contains a certain number of relative documents in the upper part. However, unless this assumption holds, it usually gives a worse ranking than the initial search. Thus several researchers propose some specific procedure to make pseudo feedback be effective [3, 4]. In another way, Onoda [5] tried to apply one-class SVM (Support Vector Machine) to relevance feedback. Their purpose is to improve search performance by using only non-relevant documents. Though their motivation is similar to ours in terms of applying a machine learning method to complement the lack of relevance information, the assumption is somewhat different. Our assumption is to utilizes manual but the minimum relevance judgment.

Transductive leaning has already been applied in the field of image retrieval [6]. In this research, they proposed a transductive method called the manifold-ranking algorithm and showed its effectiveness by comparing with active learning based Support Vector Machine. However, their setting of relevance judgment is not different from many other traditional researches. They fix the total number of images that are marked by a user to 20. As we have already claimed, this setting is not practical because most users feel that 20 is too much for judgment. We think none of research has not yet answered the question. For relevance judgment, most of the researches have adopted either of the following settings. One is the setting of "Enough relevant documents are available", and the other is "No relevant document is available". In contrast to them, we adopt the setting of "Only one relevant document is available". Our aim is to achieve performance improvement with the minimum effort of judging relevancy of documents.

The reminder of this paper is structured as follows. Section 2 and 3 describe two fundamental techniques for our query expansion method. Section 4 explains a technique to complement the smallness of manual relevance judgment. Section 5 introduces a whole procedure of our query expansion method step by step. Section 6 shows empirical evidence of the effectiveness of our method compared with two traditional query expansion methods. Section 7 investigates the experimental results more in detail. Finally, Section 8 summarizes our findings.

2 Basic Techniques

2.1 Query Expansion

The main objective of query expansion is to select additional terms of achieving better search results. From where and how to choose such terms differentiate

many query expansion techniques which have been proposed so far. For example, a method for domain specific search prepares documents in a certain domain and pick up terms from them as a batch procedure [7, 8]. In another case, a method for ad-hoc search usually selects terms from documents at the head of an initial search result. This approach further branches off to the utility of manual or automatic feedback. Anyway, most of the methods first score each term in a certain set of documents and then choose some best scored terms for expansion.

Our method belongs to the latter approach - query expansion for ad-hoc search with manual feedback. In this approach, there is a well-known query expansion method called the Robertson's *wpq* method [1] which is used in many researches [3, 4]. Our method is based on this one. The *wpq* selects expansion terms using the following scoring function.

$$score(t) = \left(\frac{r_t}{R} - \frac{n_t - r_t}{N - R}\right) * \log \frac{r_t/(R - r_t)}{(n_t - r_t)/(N - n_t - R + r_t)} \tag{1}$$

where r_t is the number of seen relevant documents containing term t. n_t is the number of documents containing t. R is the number of seen relevant documents for a query. N is the number of documents in the collection. The second term of this formula is called the Robertson/Spark Jones weight [9] which is the core of the term weighting function in the Okapi system [10]. This function is originated in the following formula.

$$score(t) = (p_t - q_t) \log \frac{p_t(1 - q_t)}{q_t(1 - p_t)} \tag{2}$$

where p_t is the probability that a term t appears in relevant documents. q_t is the probability that a term t appears in non-relevant documents. If we estimate p_t with $\frac{r_t}{R}$ and q_t with $\frac{n_t - r_t}{N - R}$, we can get fomula (1). Since the number of non-relevant documents can be prepared easily, $\frac{n_t - r_t}{N - R}$ is likely to be a good estimation for q_t. In contrast, it is not so easy to give a good estimation for p_t. Since users in practical situations do not give much feedback, R tends to be very small and this fact produces two problems. One is the lack of term variety. Candidates for expansion terms are limited. The other is the scoring ability of $\frac{r_t}{R}$. Since r_t is small if R is small, many terms come to have the same score. The challenge is to increase the number of R. Although pseudo feedback which automatically assumes top n documents as relevant is one solution, its performance heavily depends on the quality of an initial search. As we show later, pseudo feedback has limited performance.

Unlike pseudo feedback approach, our method tries to compensate for the smallness of R with a transductive learning technique, which is used to find documents possibly being relevant based on a set of training examples[4]. Because we want to consider an assumption not far from practical situations, we restrict the number of training examples to the minimum - a relevant document and a non-relevant document. Of course, manual judgment is an advantage to the pseudo

[4] documents with manual judgments

one. However this minimum information has no utility for the *wpq* method. Performance improvement depends on the accuracy of the judgment assigned by the learning to each document with no manual judgment.

2.2 Transductive Learning

Transductive learning is a machine learning technique based on the transduction which directly derives the classification labels of test data without making any approximating function from training data [11]. This learning technique is based on an assumption that two similar data are likely to have the same class label. If we can define a reasonable similarity between each element of a data set, this learning works well even if the number of training examples is small.

The learning task is defined on a data set X of n points. X consists of training data set $L = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l)$ and test data set $U = (\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, ..., \boldsymbol{x}_{l+u})$; typically $l \ll u$. The purpose of the learning is to assign a label to each element in U under the condition that the label of each element in L are given.

Recent researches about transductive learning have proposed several algorithms which are based on the solution for graph cutting problems [12–14]. According to the experimental results in [14], these algorithms do not have so much performance difference that we select an algorithm called "*Spectral Graph Transducer* (SGT)" for our query expansion method. The SGT formalizes a learning task as an optimization problem of the constrained ratiocut. By solving the relaxed problem, it produces an approximation to the original solution.

When applying it to query expansion, X corresponds to a set of top n ranked documents in a hit-list. Because the number of documents in a collection is usually too huge[5], n should be set to a moderate number. L corresponds to two documents with manual judgments, a relevant document and a non-relevant document. Furthermore, U corresponds to the documents of $X \cap \overline{L}$ whose relevancy is unknown. In the learning process, first SGT makes an undirected graph where a vertex corresponds to a document in X and an edge represents similarity between vertices. For each vertex, edges to most k similar vertices are created in the graph. The problem here is how to partition it to two parts where one part includes only positive examples (relevant documents) and the other includes only negative examples (non-relevant documents). SGT formalizes it as the following constrained ratiocut problem.

$$max_{\boldsymbol{y}} \quad \frac{cut(G^+, G^-)}{|\{i : y_i = 1\}||\{i : y_i = -1\}} \tag{3}$$

$$s.t. \quad y_i = 1, \quad if\ i \in Y_l\ and\ positive \tag{4}$$

$$y_i = -1, \quad if\ i \in Y_l\ and\ negative \tag{5}$$

$$\boldsymbol{y} \in \{+1, -1\}^n \tag{6}$$

This problem is based on a mincut problem. $cut(G^+, G^-)$ in the formula (3) represents a cut of a k-nearest graph described above. Although we can

[5] Normally it is more than ten thousand.

solve the learning task as a simple mincut problem, there is a risk that an unbalanced label assignment is produced as Joachims points out. Because such an assignment is not likely to be a good solution, SGT introduces a constraint in the denominator of the formula (3) to produce a more balanced label assignment. This new problem is hard to solve as it is, thus SGT gives an approximation to the solution by solving its relaxed problem. We omit the details about its concrete solution (See more details in [12]). At final stage of SGT, it assigns a value around $\hat{\gamma}_+ = +\sqrt{\frac{1-\hat{p}}{\hat{p}}}$ for examples possibly being positive and $\hat{\gamma}_- = +\sqrt{\frac{\hat{p}}{1-\hat{p}}}$ for examples possibly being negative. Here $\hat{p}$ is an estimate for the fraction of positive examples in X. According to Joachims, SGT has several parameters which give large influence to its learning performance. In particular, performance of our query expansion method is very sensitive to $\hat{p}$. We next explain how to relax this sensitivity.

3 A Modified Term Scoring Scheme

If we use the scoring function in the formula (1), we have to assign a binary label (1 for a relevant document and 0 for a non-relevant) to each document based on a value assigned by SGT. This means that it is necessary a certain threshold to make hard class assignment. SGT now tentatively use a threshold $\theta = \frac{\hat{\gamma}_+ + \hat{\gamma}_-}{2}$. However, $\hat{\gamma}_+$ and $\hat{\gamma}_-$ are sometimes not reliable because $\hat{p}$ is difficult to estimate in our setting. Accordingly, instead of binary labels, we use a scoring function in formula (2) with another estimation of $\hat{p}_t$. SGT finally assigns a value $z_i = \hat{\gamma}_+ - \theta$ or $\hat{\gamma}_- - \theta$ which distributes around 0 to each document d_i. If z_i is positive, the corresponding document seems to be relevant with strong possibility. Similarly, if the value is negative, the corresponding document seems to be non-relevant with strong possibility.

Our method quantifies the possibility using a simple function which assigns a real number of less than 1.0 to each example in X. Because it is important to make a loose threshold allowing examples to which SGT acutually assigns negative values, following functions are tested as representatives in our research. Figure 1 show the shape of each function.

1. Step function (SGT-step)

$$f(x) = \begin{cases} 1 & x \geq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

2. Partially Linear function (SGT-linear)

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 1 + x & -1 \leq x < 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

3. Partially normal distribution function (SGT-ndist)

$$f(x) = \begin{cases} 1 & x \geq 0 \\ \exp(-2x^2) & -1 < x < 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

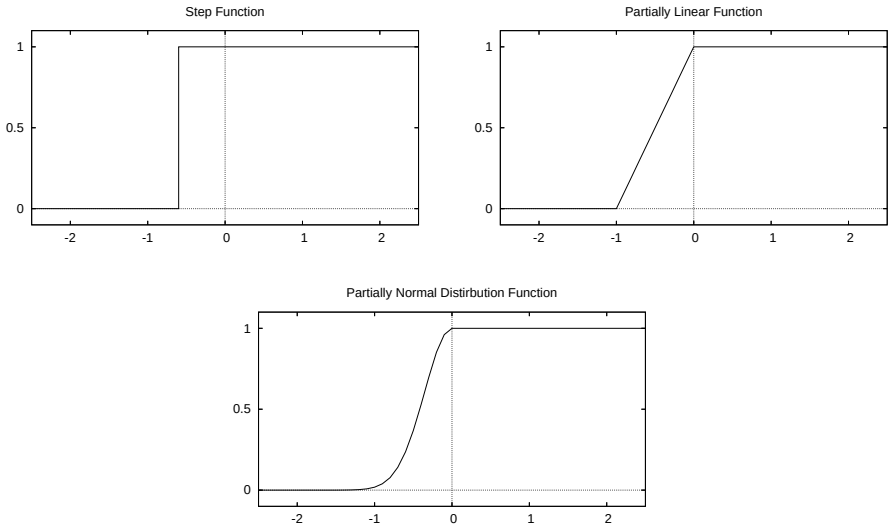

Fig. 1. Simple functions for the estimation of p_t

The first function just shifts the original threshold. The second and the third ones set a loose threshold which includes some examples assigned negative values by SGT. Using values produced by one of these functions, our method estimates p_t in the following way.

$$p_t = \sum_{i:t \in d_i} f(z_i) / \sum_{i=1}^{n} f(z_i) \tag{10}$$

The difference As described before, we estimate q_t with $\frac{n_t - r_t}{N - R}$ where R is a set of documents d_i whose z_i is positive.

4 Expansion Procedures

We here explain a whole procedure of our query expansion method step by step.

1. **Initial Search**: A retrieval starts by inputting a query for a topic to an IR system.
2. **Relevance Judgment for Documents in a Hit-List**: The IR system returns a hit-list for the initial query. Then the hit-list is scanned to check whether each document is relevant or non-relevant in descending order of the ranking. In our assumption, this reviewing process terminates when a relevant document and a non-relevant one are found.
3. **Finding more relevant documents by transductive learning**: Because only two judged documents are too few to estimate p_i and $\bar{p_i}$ correctly, our

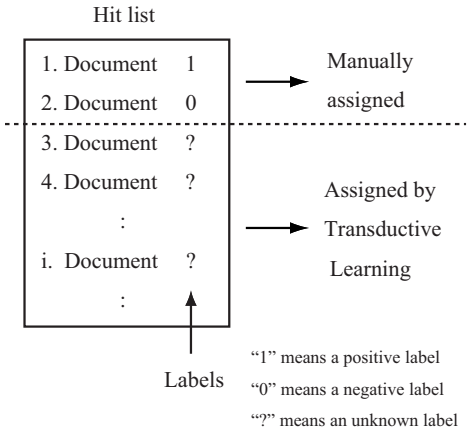

Fig. 2. Label Assignment to some Top-Ranked Documents in a Hit-List by Transductive Learning

query expansion tries to increase the number of relevant documents for the *wpq* formula using the SGT transductive learning algorithm. As shown in Figure2, SGT assigns a value of the possibility to be relevant for the topic to each document with no relevance judgment (documents under the dashed line in the Fig) based on two judged documents (documents above the dashed line in the Fig).

4. **Selecting terms to expand the initial query**: Our query expansion method calculates the score of each term appearing in relevant documents (including documents judged as relevant by SGT) using *wpq* formula, and then selects a certain number of expansion terms according to the ranking of the score. Selected terms are added to the initial query. Thus an expanded query consists of the initial terms and added terms.
5. **The Next Search with an expanded query**: The expanded query is inputted to the IR system and a new hit-list will be returned. One cycle of query expansion finishes at this step.

   In the above procedures, we naturally introduced transductive learning into query expansion as the effective way in order to automatically find some relevant documents. Thus we do not need to modify a basic query expansion procedure and can fully utilize the potential power of the basic query expansion.

   The computational cost of transductive learning is not so much. Actually transductive learning takes a few seconds to label 100 unlabeled documents and query expansion with all the labeled documents also takes a few seconds. Thus our system can expand queries sufficiently quick in practical applications.

5 Experiments

This section provides empirical evidence on how our query expansion method can improve the performance of information retrieval. We compare our method with other traditional methods.

5.1 Environmental Settings

We use the Okapi [10] as a retrieval system and a data set for the TREC-8 adhoc task [15].

We select the BM25 as a weight function in Okapi. It calculates the score of each document in a collection based on the following formula.

$$\sum_{T \in Q} w^{(1)} \cdot \frac{(k_1+1)tf(k_3+1)qtf}{(K+tf)(k_3+qtf)} \tag{11}$$

$$K = k_1 \left((1-b) + b\frac{dl}{avdl} \right) \tag{12}$$

where Q is a query containing terms T, tf is the frequency of occurrence of the term within a document, qtf is the frequency of the term within the topic from which Q was derived. K is calculated by (12), where dl and $avdl$ denote the document length and the average document length measured in some suitable unit, such as word or sequence of words. In our experiments, we set $k_1 = 1.2, k_3 = 1000, b = 0.75$, and $avdl = 135.6$. $w^{(1)}$ is the Robertson/Spark Jones weight introduced in section 2. When doing an initial search for each topic, this weight is calculated by the following formula.

$$w^{(1)} = \log \frac{N - n_t + 0.5}{n_t + 0.5} \tag{13}$$

The data set consists of a document collection, 50 topics (No.401-450) and a list of manual relevance judgment. The document collection contains about 520,000 news articles. Each document is preprocessed by removing stopwords and stemming. Query terms for an initial search are nouns extracted from the **title** tag in each topic's description. Some topics have few relevant documents or too much relevant documents. We remove such topics having none of relevant or non-relevant document within top 10 documents because we cannot apply our query expansion method for such topics. There are 8 such exceptive topics in our experiments.

5.2 Query Expansion Methods to Compare

We compared our query expansion method with the following two others.

Normal : This method simply uses only one relevant documents judged by hand. This is called *incremental relevance feedback* [16–18].

Table 1. 11 Points Average Precision

Number of terms added	5	10	15	20
Normal	0.191	0.175	0.164	0.162
Pseudo	0.213	0.210	0.206	0.206
SGT-step-0	0.230	0.230	0.220	0.215
SGT-step-α	**0.245**	0.241	0.240	0.231
SGT-linear	0.238	**0.249**	**0.257**	**0.240**
SGT-ndist	0.246	0.248	0.245	0.239

Pseudo : This method is called *pseudo relevance feedback*, which assumes top n documents as relevant ones. we set 30 for n in our experiments. 30 is the best value in our preliminary experiments.

According to the difference of term scoring scheme, we test four types of SGT-based query expansion methods. They are represented by **SGT-step-0**, **SGT-step-α**, **SGT-linear** and **SGT-ndist** respectively. **SGT-step-0** and **SGT-step-α** differs in each value of α. $\alpha = 0$ for the former, and α in the latter is the 30th largest value in all of z_i. The latter is another kind of Pseudo method mixed by SGT.

5.3 Results

We evaluated the results in two ways. One is 11 points average precision (in Table 1). The other is precision-recall curve (in Fig 3). The number of expansion terms we tested are 5,10,15 and 20. Precision and recall are calculated on residual collection where documents with manual judgment (2 documents in our case) are removed from an original collection. The value in the table is averaged over 42 topics. The number of documents used for SGT is 100. Since 2 documents are given as training examples, the rest 98 documents are used as test examples. SGT has several parameters to be set such as fraction of relevant documents, number of nearest neighbor in a graph, number of eigen values to use and so on. Although a default value or an automatic calculation procedure is prepared for each parameter, the parameter of fraction of relevant documents does not work well without manual setting because the number of training examples is too few. We set this parameter 0.1 for all topics based on our preliminary test.

Table 1 shows average precisions. All the SGT-based methods achieved higher precision compared with Normal and Pseudo methods. As for the usefuleness of functions for the estimation of p_t, SGT-step-0 is slightly less than the other three SGT-based methods. Thus we can say that the functions are effective. However any distinctive advantage could not be seen among three functions. Since the number of expansion terms did not affect retrieval performance, we only make precision-recall curves when 5 expansion terms are added as shown in Figure 3. Curves of SGT based methods did not across over other curves at any point. This indicates the superiority of our query expansion method.

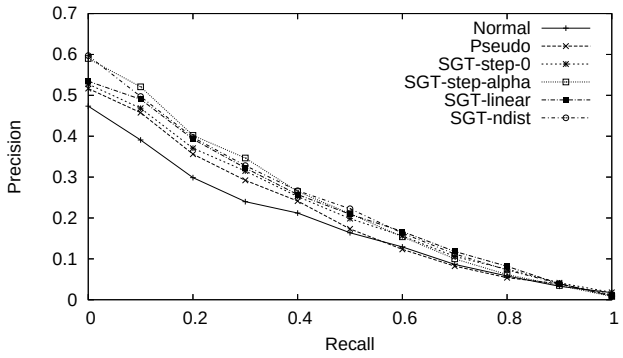

Fig. 3. Precision-Recall Curve

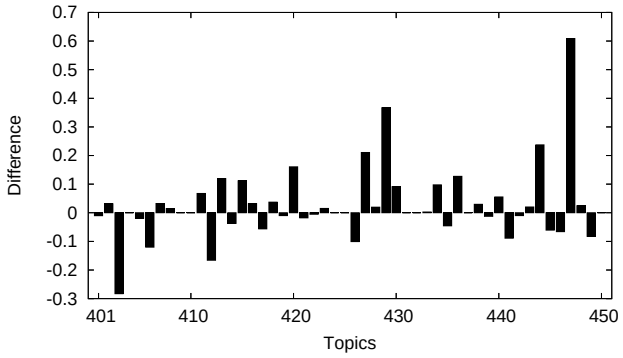

Fig. 4. Difference of 11 Points Average Precision between SGT-step-α and Pseudo

6 Discussion

Figure 4 shows a bar graph of difference of 11 points average precision between SGT-step-α and Pseudo for each topic when 5 expansion terms are added. SGT-step-α is superior to Pseudo if the difference is positive, and vice versa. As shown in the figure, there are some topics for which SGT-step-α gives worse performance than Pseudo. Because this result is related to the number of relevant documents used for query expansion, we also investigated a difference of the number of relevant documents used for query expansion in Figure 5. When SGT-step-α has less relevant documents than Pseudo, the performance is also less than Pseudo except for some topics. The reason SGT-step-α has less relevant documents than Pseudo is a miss setting of the fraction of relevant documents which is a parameter of SGT. It is better to set a correct value for the fraction of relevant documents to SGT, however estimating the value is not so easy.

We have investigated that our query expansion methods are superior to other traditional methods from several points of view. However we have another ques-

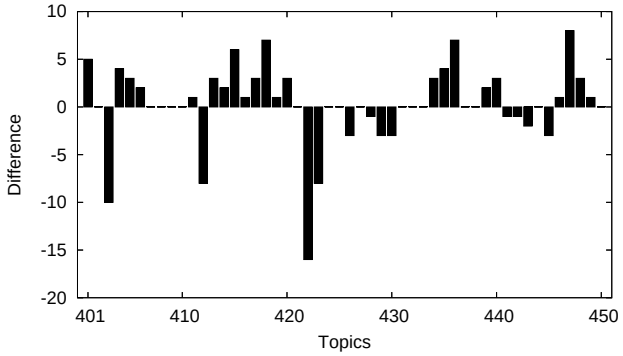

Fig. 5. Difference of the number of relevant documents in top 30 between SGT-step-α and Pseudo

Table 2. Recall at nth rank

n	SGT only	SGT-based QE
10	0.042	0.129
20	0.083	0.185
30	0.117	0.226
40	0.155	0.257
50	0.188	0.282
60	0.214	0.294
70	0.258	0.314
80	0.297	0.331
90	0.309	0.345
100	0.346	0.363

tion that query expansion procedure is necessary because SGT can re-order documents by itself. In order to answer this question, we compare recall of top n ($n = 10 \sim 100$) documents in a hit-list re-ordered by SGT itself and SGT-based query expansion method (SGT-step-α) as shown in Table 2. As shown in the table, recall rates of SGT itself is lower than the SGT-based query expansion method. This shows that SGT itself cannot work well. Under the condition of the minimum relevance judgment, SGT is effective if being mixed with our specific procedure.

7 Conclusion

In this paper we proposed a novel query expansion method which only use the minimum manual judgment. To complement the lack of relevant documents, this method utilizes the SGT transductive learning algorithm to predict the relevancy of unjudged documents. Since the performance of SGT much depends on an estimation of the fraction of relevant documents, we introduced a modified term

scoring scheme which actually changes the thresholding procedure of SGT. The experimental results showed our method outperforms other traditional methods in the evaluations of precision and recall criteria. Though our modified term scoring scheme could relax SGT's parameter sensitivity described above in some degree, we have more chance to improve the performance by removing more SGT's parameter dependencies.

References

1. I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In *Proceedings of SIGIR 2003*, pages 213–220, 2003.
2. S. Dumais and et al. Sigir 2003 workshop report: Implicit measures of user interests and preferences. In *SIGIR Forum*, 2003.
3. S. Yu and et al. Improving pseud-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of WWW 2003*, 2003.
4. A. M. Lam-Adesina and G. J. F. Jones. Applying summarization techniques for term selection in relevance feedback. In *Proceedings of SIGIR 2001*, pages 1–9, 2001.
5. T. Onoda, H. Murata, and S. Yamada. Non-relevance feedback document retrieva. In *Proceedings of CIS 2004*. IEEE, 2003.
6. J. He and et al. Manifold-ranking based image retrieval. In *Proceedings of Multimedia 2004*, pages 9–13. ACM, 2004.
7. G. W. Flake and et al. Extracting query modification from nonlinear svms. In *Proceedings of WWW 2002*, 2002.
8. S. Oyama and et al. keysword spices: A new method for building domain-specific web search engines. In *Proceedings of IJCAI 2001*, 2001.
9. S. E. Robertson. On term selection for query expansion. *Journal of Documentation*, 46(4):359–364, 1990.
10. S. E. Robertson. Overview of the okapi projects. *Journal of the American Society for Information Science*, 53(1):3–7, 1997.
11. V Vapnik. *Statistical learning theory*. Wiley, 1998.
12. T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of ICML 2003*, pages 143–151, 2003.
13. X Zhu and et al. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML 2003*, pages 912–914, 2003.
14. A. Blum and et al. Semi-supervised learning using randomized mincuts. In *Proceedings of ICML 2004*, 2004.
15. E. Voorhees and D. Harman. Overview of the eighth text retrieval conference. 1999.
16. I. J. Aalbersberg. Incremental relevance feedback. In *Proceedings of SIGIR '92*, pages 11–22, 1992.
17. J. Allan. Incremental relevance feedback for information filtering. In *Proceedings of SIGIR '96*, pages 270–278, 1996.
18. M. Iwayama. Relevance feedback with a small number of relevance judgements: Incremental relevance feedback vs. document clustering. In *Proceedings of SIGIR 2000*, pages 10–16, 2000.