Query Expansion with the Minimum User Feedback by Transductive Learning

Masayuki OKABE
Information and Media Center
Toyohashi University of Technology
Aichi, 441-8580, Japan
okabe@imc.tut.ac.jp

Kyoji UMEMURA
Information and Computer Sciences
Toyohashi University of Technology
Aichi, 441-8580, Japan
umemura@tutics.tut.ac.jp

Seiji YAMADA
National Institute of Informatics
Tokyo,101-8430, 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 application of a transductive learning technique in order to increase relevant documents. The other is a modified parameter estimation method which laps the predictions by multiple learning trials and try to differentiate the importance of candidate terms for expansion in relevant documents. Experimental results show that our technique outperforms some traditional query expansion methods in several evaluation measures.

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 (Ruthven, 2003). 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 (Dumais and et al., 2003).

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. When applying the learning, there occurs some difficult problems of parameter settings. We also try to provide a reasonable resolution for the problems and show the effectiveness of our proposed method in experiments.

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-

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 963–970, Vancouver, October 2005.

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 (Yu and et al, 2003; Lam-Adesina and Jones, 2001). In another way, Onoda (Onoda et al., 2004) 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 (He and et al., 2004). 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 describes two fundamental techniques for our query expansion method. Section 3 explains a technique to complement the smallness of manual relevance judgment. Section 4 introduces a whole procedure of our query expansion method step by step. Section 5 shows empirical evidence of the effectiveness of our method compared with two traditional query expansion methods. Section 6 investigates the experimental results more in detail. Finally, Section 7 summarizes our findings.

2 Basic Methods

2.1 Query Expansion

So far, many query expansion techniques have been proposed. While some techniques focus on the domain specific search which prepares expansion terms in advance using some domain specific training documents (Flake and et al, 2002; Oyama and et al, 2001), most of techniques are based on relevance feedback which is given automatically or manually.

In this technique, expansion terms are selected from relevant documents by a scoring function. The Robertson’s *wpq* method (Ruthven, 2003) is often used as such a scoring function in many researches (Yu and et al, 2003; Lam-Adesina and Jones, 2001). We also use it as our basic scoring function. It calculates the score of each term by the following formula.

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

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 (Robertson, 1990) which is the core of the term weighting function in the Okapi system (Robertson, 1997).

This formula is originated in the following formula.

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

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. We can easily notice that it is very important how the two probability of p_t and q_t should be estimated. The first formula estimates p_t with $\frac{r_t}{R}$ and q_t with $\frac{N_t - R_t}{N - R}$. For the good estimation of p_t and q_t, plenty of relevant document is necessary. Although pseudo feedback which automatically assumes top n documents as relevant is one method and is often used, its performance heavily depends on the quality of an initial search. As we show later, pseudo feedback has limited performance.

We here consider a query expansion technique which uses manual feedback. It is no wonder manual feedback shows excellent and stable performance if enough relevant documents are available, hence the challenge is how it keeps high performance with less amount of manual relevance judgment. In particular, we restrict the manual judgment to the minimum amount, namely only **a relevant document and a non-relevant document**. In this assumption, the problem is how to find more relevant documents based on a relevant document and a non-relevant document. We use transductive learning technique which is suitable for the learning problem where there is small training examples.

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 (Vapnik, 1998). Because it does not need to make approximating function, it works well even if the number of training data is small.

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

Recently, transductive learning or semi-supervised learning is becoming an attractive subject in the machine learning field. Several algorithms have been proposed so far (Joachims, 1999; Zhu and et al., 2003; Blum and et al., 2004) and they show the advantage of this approach in various learning tasks. In order to apply transductive learning to our query expansion, we select an algorithm called "*Spectral Graph Transducer* (SGT)“ (Joachims, 2003), which is one of the state of the art and the best transductive learning algorithms. SGT formalizes the problem of assigning labels to U with an optimization problem of the constrained ratiocut. By solving the relaxed problem, it produces an approximation to the original solution.

When applying SGT to query expansion, X corresponds to a set of top n ranked documents in a hit-list. X does not corresponds to a whole document collection because the number of documents in a collection is too huge[1] for any learning system to process. 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. SGT is used to produce the relevancy of documents in U. SGT actually assigns values around $\gamma_+ - \theta$ for documents possibly being relevant and $\gamma_- - \theta$ for documents possibly being non-relevant. $\gamma_+ = +\sqrt{\frac{1-f_p}{f_p}}$, $\gamma_- = -\sqrt{\frac{f_p}{1-f_p}}$, $\theta = \frac{1}{2}(\gamma_+ + \gamma_-)$, and f_p is the fraction of relevant documents in X. We cannot know the true value of f_p in advance, thus we have to estimate its approximation value before applying SGT.

According to Joachims, parameter k (the number of k-nearest points of a data $\vec{x}$) and d (the number of eigen values to ...) give large influence to SGT's learning performance. Of course those two parameters should be set carefully. However, besides them, f_p is much more important for our task because it controls the learning performance. Since extremely small L (actually $|L| = 2$ is our setting) give no information to estimate the true value of f_p, we do not strain to estimate its single approximation value but propose a new method to utilize the results of learning with some promising f_p. We describe the method in the next section.

3 Parameter Estimations based on Multiple SGT Predictions

3.1 Sampling for Fraction of Positive Examples

SGT prepares 2 estimation methods to set f_p automatically. One is to estimate from the fraction of positive examples in training examples. This method is not suitable for our task because f_p is always fixed to 0.5 by this method if the number of training examples changes despite the number of relevant documents is small in many practical situations. The other is to estimate with a heuristic that the difference between a setting of f_p and the fraction of positive examples actually assigned by SGT should be as small as possible. The procedure provided by SGT starts from $f_p = 0.5$ and the next f_p is set to the fraction of documents assigned as relevant in the previous SGT trial. It repeats until f_p changes

[1] Normally it is more than ten thousand.

```
Input
  $N_{tr}$   // the number of training examples
Output
  $S$        // a set of sampling points

$piv = \ln(N_{tr})$;  // sampling interval
$nsp = 0$;            // the number of sampling points
for($i = piv$; $i \leq N_{tr} - 1$; $i+ = piv$){
    add $i$ to ;
    nsp++;
    if(nsp == 10){ exit; }
}
```

Figure 1: Pseudo code of sampling procedure for f_p

five times or the difference converges less than 0.01. This method is neither works well because the convergence is not guaranteed at all.

Presetting of f_p is primarily very difficult problem and consequently we take another approach which laps the predictions of multiple SGT trials with some sampled f_p instead of setting a single f_p. This approach leads to represent a relevant document by not a binary value but a real value between 0 and 1. The sampling procedure for f_p is illustrated in Figure 1. In this procedure, sampling interval changes according to the number of training examples. In our preliminary test, the number of sampling points should be around 10. However this number is adhoc one, thus we may need another value for another corpus.

3.2 Modified estimations for p_t and q_t

Once we get a set of sampling points $S = \{f_p^i : i = 1 \sim 10\}$, we run SGT with each f_p^i and laps each resultant of prediction to calculate p_t and q_t as follows.

$$p_t = \frac{\sum_i r_t^i}{\sum_i R_i} \tag{3}$$

$$q_t = \frac{\sum_i n_t - r_t^i}{\sum_i N - R_i} \tag{4}$$

Here, R_i is the number of documents which SGT predicts as relevant with ith value of f_p^i, and r_t^i is the number of documents in R_i where a term t appears. In each trial, SGT predicts the relevancy of documents by binary value of 1 (for relevant) and 0 (for non-relevant), yet by lapping multiple resultant of predictions, the binary prediction value changes to a real value which can represents the relevancy of documents in more detail. The main merit of this approach in comparison with fixing f_p to a single value, it can differentiate a value of p_t if N_{tr} is small.

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_t and q_t correctly, our 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 Figure).

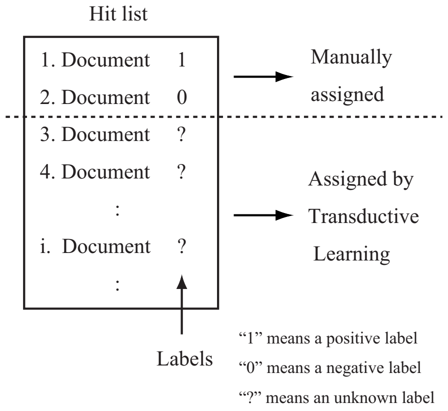
Hit list
1. Document 1
2. Document 0
3. Document ?
4. Document ?
:
i. Document ?
:
Manually assigned
Assigned by Transductive Learning
Labels
"1" means a positive label
"0" means a negative label
"?" means an unknown label

Figure 2: A method to find tentative relevant documents

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

5.1.1 Data set

We use the TREC-8 data set (Voorhees and Harman, 1999) for our experiment. The document corpus contains about 520,000 news articles. Each document is preprocessed by removing stopwords and stemming. We also use fifty topics (No.401-450) and relevance judgments which are prepared for adhoc task in the TREC-8. Queries for an initial search are nouns extracted from the **title** tag in each topic.

5.1.2 Retrieval Models

We use two representative retrieval models which are bases of the Okapi (Robertson, 1997) and SMART systems. They showed highest performance in the TREC-8 competition.

Okapi : The weight function in Okapi is BM25. It calculates each document's score by the following formula.

$$score(d) = \sum_{T \in Q} w^{(1)} \cdot \frac{(k_1+1)tf(k_3+1)qtf}{(K+tf)(k_3+qtf)} \quad (5)$$

$$w^{(1)} = \log \frac{(r_t+0.5)/(R-r_t+0.5)}{(n_t-r_t+0.5)/(N-n_t-R+r_t+0.5)} \quad (6)$$

$$K = k_1\left((1-b)+b\frac{dl}{avdl}\right) \quad (7)$$

where Q is a query containing terms T, tf is the term's frequency in a document, qtf is the term's frequency in a text from which Q was derived. r_t and n_t are described in section 2. K is calculated by (7), where dl and $avdl$ denote the document length and the average document length. In our experiments, we set $k_1 = 1.2, k_3 = 1000, b = 0.75$, and $avdl = 135.6$. Terms for query expansion are ranked in decreasing order of $r_t \times w^{(1)}$ for the following Okapi's retrieval tests without SGT (**Okapi_manual** and **Okapi_pseudo**) to make conditions the same as of TREC-8.

SMART : The SMART's weighting function is as follows[2].

$$score(d) = \sum_{T \in Q} \{1+\ln(1+\ln(tf))\} * \log(\frac{N+1}{df}) * pivot \quad (8)$$

$$pivot = \frac{1}{0.8+0.2 \times \frac{dl}{avdl}} \quad (9)$$

df is the term's document frequency. tf, dl and $avdl$ are the same as Okapi. When doing relevance feedback, a query vector is modified by the following Rocchio's method (with parameters $\alpha = 3$, $\beta = 2$, $\gamma = 2$).

$$\vec{Q}_{new} = \alpha\vec{Q}_{old} + \frac{\beta}{|D_{rel}|}\sum_{D_{rel}}\vec{d} - \frac{\gamma}{|D_{nrel}|}\sum_{D_{nrel}}\vec{d} \quad (10)$$

[2] In this paper, we use AT&T's method (Singhal et al., 1999) applied in TREC-8

Table 1: Results of Initial Search

	P10	P30	RPREC	MAP	R05P
Okapi_ini	0.466	0.345	0.286	0.239	0.195
SMART_ini	0.460	0.336	0.271	0.229	0.187

D_{rel} and D_{nrel} are sets of seen relevant and non-relevant documents respectively. Terms for query expansion are ranked in decreasing order of the above Rocchio's formula.

Table 1 shows their initial search results of Okapi (**Okapi_ini**) and SMART (**SMART_ini**). We adopt five evaluation measures. Their meanings are as follows (Voorhees and Harman, 1999).

P10 : The precision after the first 10 documents are retrieved.

P30 : The precision after the first 30 documents are retrieved.

R-Prec : The precision after the first R documents are retrieved, where R is the number of relevant documents for the current topic.

MAP : Mean average precision (MAP) is the average precision for a single topic is the mean of the precision obtained after each relevant document is retrieved (using zero as the precision for relevant documents that are not retrieved).

R05P : Recall at the rank where precision first dips below 0.5 (after at least 10 documents have been retrieved).

The performance of query expansion or relevance feedback is usually evaluated on a residual collection where seen documents are removed. However we compare our method with pseudo feedback based ones, thus we do not use residual collection in the following experiments.

5.1.3 Settings of Manual Feedback

For manual feedback, we set an assumption that a user tries to find relevant and non-relevant documents within only top 10 documents in the result of an initial search. If a topic has no relevant document or no non-relevant document in the top 10 documents, we do not apply manual feedback, instead we consider the result of the initial search for such topics. There are 8 topics [3] which we do not apply manual feedback methods.

Table 2: Results of **Okapi_sgt** (5 terms expanded)

	P10	P30	RPREC	MAP	R05P
20	0.516	0.381	0.308	0.277	0.233
50	0.494	0.380	0.286	0.265	0.207
100	0.436	0.345	0.283	0.253	0.177

Table 3: Results of **Okapi_sgt** (10 terms expanded)

	P10	P30	RPREC	MAP	R05P
20	0.508	0.383	0.301	0.271	0.216
50	0.520	0.387	0.294	0.273	0.208
100	0.494	0.365	0.283	0.261	0.190

Table 4: Results of **Okapi_sgt** (15 terms expanded)

	P10	P30	RPREC	MAP	R05P
20	0.538	0.381	0.298	0.274	0.223
50	0.528	0.387	0.298	0.283	0.222
100	0.498	0.363	0.280	0.259	0.197

Table 5: Results of **Okapi_sgt** (20 terms expanded)

	P10	P30	RPREC	MAP	R05P
20	0.546	0.387	0.307	0.289	0.235
50	0.520	0.385	0.299	0.282	0.228
100	0.498	0.369	0.272	0.255	0.188

5.2 Basic Performance

Firstly, we evaluate the basic performance of our query expansion method by changing the number of training examples. Since our method is based on Okapi model, we represent it as **Okapi_sgt** (with parameters $k = 0.5 * N_{tr}$, $d = 0.8 * N_{tr}$. k is the number of nearest neighbors, d is the number of eigen values to use and N_{tr} is the number of training examples).

Table 2-5 shows five evaluation measures of **Okapi_sgt** when the number of expansion terms changes. We test 20, 50 and 100 as the number of training examples and 5, 10 15 and 20 for the number of expansion terms. As for the number of training examples, performance of 20 and 50 does not differ so much in all the number of expansion terms. However performance of 100 is clearly worse than of 20 and 50. The number of expansion terms does not effect so much in every evaluation measures. In the following experiments, we compare the results of **Okapi_sgt** when the number of training examples is 50 with other query expansion methods.

[3] Topic numbers are 409, 410, 424, 425, 431, 432, 437 and 450

Table 6: Results of Manual Feedback Methods (MAP)

	5	10	15	20
Okapi_sgt	0.265	0.273	0.274	0.282
Okapi_man	0.210	0.189	0.172	0.169
SMART_man	0.209	0.222	0.220	0.219

Table 7: Results of Manual Feedback Methods (10 terms expanded)

	P10	P30	RPREC	MAP	R05P
Okapi_sgt	0.520	0.387	0.294	0.273	0.208
Okapi_man	0.420	0.285	0.212	0.189	0.132
SMART_man	0.434	0.309	0.250	0.222	0.174

5.3 Comparison with other Manual Feedback Methods

We next compare our query expansion method with the following manual feedback methods.

Okapi_man : This method simply uses only one relevant document judged by hand. This is called *incremental relevance feedback* (Aalbersberg, 1992; Allan, 1996; Iwayama, 2000).

SMART_man : This method is SMART's manual relevance feedback (with parameters $\alpha = 3$, $\beta = 2, \gamma = 0$). γ is set to 0 because the performance is terrible if γ is set to 2.

Table 6 shows the mean average precision of three methods when the number of expansion terms changes. Since the number of feedback documents is extremely small, two methods except for **Okapi_sgt** get worse than their initial searches. **Okapi_man** slightly decreases as the number of expansion terms increases. Contrary, **SMART_man** do not change so much as the number of expansion terms increases. Table 7 shows another evaluation measures with 10 terms expanded. It is clear that **Okapi_sgt** outperforms the other two methods.

5.4 Comparison with Pseudo Feedback Methods

We finally compare our query expansion method with the following pseudo feedback methods.

Okapi_pse : This is a pseudo version of Okapi which assumes top 10 documents in the initial search as relevant ones as well as TREC-8 settings.

Table 8: Results of Pseudo Feedback Methods (MAP)

	5	10	15	20
Okapi_sgt	0.265	0.273	0.274	0.282
Okapi_pse	0.253	0.249	0.247	0.246
SMART_pse	0.236	0.243	0.242	0.242

Table 9: Results of Pseudo Feedback Methods (10 terms expanded)

	P10	P30	RPREC	MAP	R05P
Okapi_sgt	0.520	0.387	0.294	0.273	0.208
Okapi_pse	0.478	0.369	0.279	0.249	0.206
SMART_pse	0.466	0.359	0.272	0.243	0.187

SMART_pse : This is a pseudo version of SMART. It also assumes top 10 documents as relevant ones. In addition, it assumes top 500-1000 documents as non-relevant ones.

In TREC-8, above two methods uses TREC1-5 disks for query expansion and a phase extraction technique. However we do not adopt these methods in our experiments[4]. Since these methods showed the highest performance in the TREC-8 adhoc task, it is reasonable to compare our method with them as competitors.

Table 8 shows the mean average precision of three methods when the number of expansion terms changes. Performance does not differ so much if the number of expansion terms changes. **Okapi_sgt** outperforms at any number of expansion. Table 9 shows the results in other evaluation measures. **Okapi_sgt** also outperforms except for **R05P**. In particular, performance in **P10** is quite well. It is preferable behavior for the use in practical situations.

6 Discussion

In the experiments, the feedback documents for **Okapi_sgt** is top ranked ones. However some users do not select such documents. They may choose another relevant and non-relevant documents which rank in top 10. Thus we test an another experiment where relevant and non-relevant documents are selected randomly from top 10 rank. Table 10 shows the result. Compared with table 2, the performance seems to become slightly worse. This shows that a

[4] Thus the performance in our experiments is a bit worse than the result of TREC-8

Table 10: Results of **Okapi_sgt** with random feedback (5 terms expanded)

	P10	P30	RPREC	MAP	R05P
20	0.498	0.372	0.288	0.265	0.222
50	0.456	0.359	0.294	0.268	0.200
100	0.452	0.335	0.270	0.246	0.186

user should select higher ranked documents for relevance feedback.

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 propose a method to sample some good fraction values. We also propose a method to laps the predictions of multiple SGT trials with above sampled fraction values and try to differentiate the importance of candidate terms for expansion in relevant documents. The experimental results showed our method outperforms other query expansion methods in the evaluations of several criteria.

References

I. J. Aalbersberg. 1992. Incremental relevance feedback. In *Proceedings of SIGIR '92*, pages 11–22.

J. Allan. 1996. Incremental relevance feedback for information filtering. In *Proceedings of SIGIR '96*, pages 270–278.

A. Blum and et al. 2004. Semi-supervised learning using randomized mincuts. In *Proceedings of ICML 2004*.

S. Dumais and et al. 2003. Sigir 2003 workshop report: Implicit measures of user interests and preferences. In *SIGIR Forum*.

G. W. Flake and et al. 2002. Extracting query modification from nonlinear svms. In *Proceedings of WWW 2002*.

J. He and et al. 2004. Manifold-ranking based image retrieval. In *Proceedings of Multimedia 2004*, pages 9–13. ACM.

M. Iwayama. 2000. Relevance feedback with a small number of relevance judgements: Incremental relevance feedback vs. document clustering. In *Proceedings of SIGIR 2000*, pages 10–16.

T. Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of ICML '99*.

T. Joachims. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of ICML 2003*, pages 143–151.

A. M. Lam-Adesina and G. J. F. Jones. 2001. Applying summarization techniques for term selection in relevance feedback. In *Proceedings of SIGIR 2001*, pages 1–9.

T. Onoda, H. Murata, and S. Yamada. 2004. Non-relevance feedback document retrieva. In *Proceedings of CIS 2004*. IEEE.

S. Oyama and et al. 2001. keysword spices: A new method for building domain-specific web search engines. In *Proceedings of IJCAI 2001*.

S. E. Robertson. 1990. On term selection for query expansion. *Journal of Documentation*, 46(4):359–364.

S. E. Robertson. 1997. Overview of the okapi projects. *Journal of the American Society for Information Science*, 53(1):3–7.

I. Ruthven. 2003. Re-examining the potential effectiveness of interactive query expansion. In *Proceedings of SIGIR 2003*, pages 213–220.

A. Singhal, S. Abney, B. Bacchiani, M. Collins, D. Hindle, and F. Pereira. 1999. At&t at trec-8.

V Vapnik. 1998. *Statistical learning theory*. Wiley.

E. Voorhees and D. Harman. 1999. Overview of the eighth text retrieval conference.

S. Yu and et al. 2003. Improving pseud-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of WWW 2003*.

X Zhu and et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML 2003*, pages 912–914.