Query expansion with naive bayes for searching distributed collections

Conference or Workshop Item

Query Expansion with Naive Bayes for Searching Distributed Collections

Hui Yang and Minjie Zhang
School of Information Technology & Computer Science
University of Wollongong
Wollongong, NSW 2522, Australia
hy92@uow.edu.au minjie@uow.edu.au

Abstract

The proliferation of online information resources increases the importance of effective and efficient distributed searching. However, the problem of word mismatch seriously hurts the effectiveness of distributed information retrieval. Automatic query expansion has been suggested as a technique for dealing with the fundamental issue of word mismatch. In this paper, we propose a method - query expansion with *Naive Bayes* to address the problem, discuss its implementation in ***IISS*** system, and present experimental results demonstrating its effectiveness. Such technique not only enhances the discriminatory power of typical queries for choosing the right collections but also hence significantly improves retrieval results.

Keywords: Naive Bayes, Query Expansion, Word Mismatch, Distributed Information Retrieval

1 Introduction

With the rapid growth of the Internet, especially World Wide Web, more and more information sources have become available online and heterogeneously distributed over the Internet. The need to search multiple collections in a distributed environment has been becoming an increasingly important problem commonly known as the resource discovery problem [12]. Searching a distributed collection presents a number of unique problems which include ranking document collections for relevance to a query, selecting the best set of collections from a ranking list, and merging the documents rankings that are returned from a set of collection.

In recent works, a number of different approaches for distributed information retrieval have been proposed and individually evaluated so as to efficiently and effectively organize, represent and search distributed collections. However, the effectiveness of searching a large set of distributed collections is significantly worse than that of searching a single centralized collection. The primary cause is that typical queries, thought adequate for document retrieval, are not very suitable for collection selection. The problem of word mismatch has seriously hurt the effectiveness of distributed information retrieval. An obvious approach to solve this problem is query expansion. When the query is expanded using words or phrases that are more specific than the words in the original query, the expanded query will therefore be more suitable for collection selection.

The primary concern of this paper is the retrieval effectiveness of distributed information retrieval based on query expansion with Naive Bayes. When a query is posted, the system firstly takes the user's original query terms as representatives of the concepts in which the user is interested. It automatically expands the terms with a *Naive Bayes* classifier using a class hierarchy with a set of labeled documents in the online thesaurus, and adds those terms that are most similar to the concept of the query to enrich the representation of the query. A complete discussion of this system, called as ***IISS*** system, can be found in [17]. We hope that this kind of query expansion results in a notable improvement in the retrieval effectiveness for searching distributed collections.

The remainder of the paper is organized as follows. In the next section, we describe related work on searching distributed collections. In Section 3, we introduce an effective query expansion technique based on *Naive Bayes* and discuss its implementation in detail in ***IISS*** system. Section 4 describes the sets of collections used for evaluation and how experiments were carried out. Section 5 presents experimental results and provides detailed analysis. Finally, we offer concluding remarks and outline future work of this research in Section 6.

2 Related Work

There have been a number of studies concerning retrieval effectiveness in a distributed environment. Xu and Croft [15] proposed a cluster – based language model in which document clustering was used to organize collections around topics, and language modeling was used to properly represent topics and effectively select the right topics for a query.

Voorhees, et, al [14] exploited the similarity of a new query to previously evaluated training queries and made use of relevant judgement from previous queries to compute the number of documents to retrieve from each collection.

Callan, et, al [2] presented that ranking collections could be addressed by an inference network in which the leaves presented document collections, and the representation nodes presented the terms that occur in the collection. The probabilities could be based upon statistics that were analogous to *tf* and *idf* in normal document retrieval, where *tf* and *idf* were always used to indicate the effectiveness of retrieval in information retrieval. The effectiveness of this approach was evaluated using the INQUERY retrieval system [1].

Gravana, et, al [7] used document frequent information of each individual collection to estimate the result size of a query in each collection and select a set of most relevant collections with these estimates.

Fuhr [5] developed a decision – theoretic model and discussed different parameters for each database: expected retrieval quality, expected number of relevant documents in the database, and cost factors for query processing and document delivery. He gave a divide – and – conquer algorithm to compute the overall optimum in order to receive the maximum number of documents at a minimum cost.

Yuwona and Lee [16] described a centralized broker architecture in which the broker maintained *df* [13] table for servers from the user query which best discriminated between servers, and then servers with higher *df* values for those terms were selected to process the query.

Frech, et, al [3, 5, 6] evaluated three of these approaches, CORI [2], CVV [16] and GLOSS [7] in a common environment and they found that there was significant room for improvement in all approaches, especially when very few information source were selected.

3 Query expansion with Naive Bayes

Most often, users have difficulties in formulating a request because they are unfamiliar with the contents of information sources, so their queries usually are very short. Such a short query tends to be inexact and ambiguous. To assist the user, ***IISS*** system attempts to provide a *conceptual retrieval* method, namely, *query expansion,* which can automatically expand the user's queries from an online thesaurus which stores word relationships. Such query expansion discovers related terms or concepts, along with their relationships with those in the user's query. Query expansion does not change the underlying information need, but makes the expanded query more suitable for information source selection.

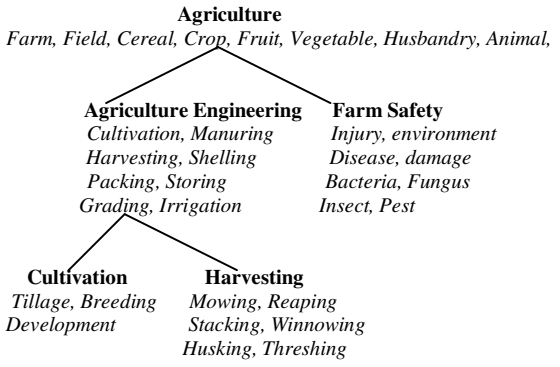

Figure 1: A subset of the topic hierarchy of online thesaurus
(Each node contains its title and the most probable keywords in italics calculated by *Naive Bayes* with training documents)

In this paper, a query expansion method is provided by *Naive Baye,* an established text classification algorithm [9,10] based on Bayesian machine learning technique.

An online thesaurus is constructed by a class hierarchy with a set of labeled training documents. Topic hierarchies are an efficient way to organize and manage a large quantity of information that would otherwise be cumbersome. Knowledge about each topic class of interests is provided in the form of its title, and some most probable keywords, as calculated by *Naïve Bayes* with a set of labeled training documents (see figure 1). The US Patent database, Yahoo and the Dewey Decimal System are all examples of topic hierarchies that exist to make information more manageable.

We consider a user's query to be associated with a *pseudo-document* indicated by *PD(Q).* The content of the pseudo-document is a list of words with the weight that occur in the preprocessed query, which can be defined as

$$PD(Q) = \{\langle t_i, w_i \rangle\} \quad (1)$$

where, t_i be terms (words) occurring in the user's query Q , and w_i be the term weight of the corresponding term in *Q*.

We then build an improved classifier by using the labeled training documents and the pseudo-document to bootstrap a Naive Bayes text classifier. This enhanced Naive Bayes classifier is used to discover new keywords that are probabilistically correlated with the original keywords in the pseudo-document.

These most probable keywords are ranked by the frequency that they occur in the training documents. Those top – ranking keywords from the same class as the pseudo-document *PD(Q)* will be added to the query and weighted appropriately. Terms in the original query are weighted more heavily than those terms which are not in the original query.

3.1.1 The *Naive Bayes* framework

We use the framework of multinominal *Naive Bayes* text classification [9,10]. The classifier parameterizes each class separately with a document frequency, and also word frequencies. Each class, c_j, has a document frequency relative to all other classes, written $P(c_k)$. For every word, w_t, in the vocabulary, *V*, $P(w_t|c_j)$ indicates the frequency that the classifier expects word w_t to occur in documents in class c_j.

Acquisition of these parameters is accomplished by using a set of labeled training documents, *D*. To estimate the word probability parameters $P(w_t|c_j)$, we count the

frequency of a word w_t which occurs among all word occurrences for documents in class c_j. Then, the estimate of the probability of word w_t in class c_j is:

$$P(w_t|c_j) = \frac{1 + \sum_{d_i \in D} N(w_t, d_i) P(c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{d_i \in D} N(w_s, d_i) P(c_j|d_i)} \quad (2)$$

where, $N(w_t, d_i)$ is the number of times that word w_t occurs in document d_i; $P(c_j|d_i) \in \{0,1\}$, is given by the labeled training documents' class label; the vocabulary *V*, $V = \{w_1, w_2, \cdots, w_{|V|}\}$; $|V|$ is the number of all words occurring in documents in class c_j.

The estimate of the class prior parameters $P(c_j)$ is set in the same way:

$$P(c_j) = \frac{1 + \sum_{d_i \in D} P(c_j|d_i)}{|C| + |D|} \quad (3)$$

where, $|C|$ indicates the number of classes, $c_j \in C = \{c_1, c_2, \cdots, c_{|C|}\}$; and $|D|$ is the number of the labeled training documents, $D = \{d_1, d_2, \cdots, d_{|D|}\}$.

3.1.2 Query expansion with a Naive Bayes classifier

Given an unlabeled document – a *pseudo-document PD(Q)* and a *Naive Bayes* classifier with the parameters $P(c_j)$ and $P(w_t|c_j)$ calculated from the labeled training documents, we can determine the probability that *PD(Q)* belongs to the class c_j using Bayes' rule and *Naive Baye*s assumption – that the probability of each word event in a document is independent of the word's context and position in the document.

$$P(c_j|d_{PD}) = \frac{P(c_j) P(d_{PD}|c_j)}{P(d_{PD})} \quad (4)$$

where, (1) $P(d_{PD}|c_j)$ is the probability of a document given its class:

$$P(d_{PD} \mid c_j) = \prod_{t}^{|d_{PD}|} (\delta + P(w_t \mid c_j)) \quad (5)$$

w_t is a word that occurs in a *pseudo-document PD(Q)*. If w_t also occurs among all word occurrences for documents in class c_j, $P(w_t|c_j)$ can be got from equation 2. But if w_t does not exist in class c_j, $P(w_t|c_j)$ is probably zero value. So we set a parameter δ, which is a very small value in the area (0, 1) in order to avoid zero value in multiplication. (2) $P(d_i)$ is the probability of a document over all mixture classes *C*, $C = \{c_1, c_2, \cdots, c_{|C|}\}$.

$$P(d_{PD}) = \sum_{r=1}^{|C|} P(c_r) P(d_{PD}|c_r) \quad (6)$$

So, we can calculate the posterior probability of each class given the evidence of the unlabeled document *PD(Q)*.

$$P(c_j|d_{PD}) = \frac{P(c_j) P(d_{PD}|c_j)}{P(d_{PD})} = \frac{P(c_j) P(d_{PD}|c_j)}{\sum_{r=1}^{|C|} P(c_r) P(d_{PD}|c_r)}$$

$$= \frac{P(c_j) \prod_{t=1}^{|d_{PD}|} (\delta + P(w_t|c_j))}{\sum_{r=1}^{|C|} \left(P(c_r) \prod_{t=1}^{|d_{PD}|} (\delta + P(w_i|c_r)) \right)} \quad (7)$$

Finally, we select the class with the highest probability that most probably contains the words with similar meaning to those in a query.

Some keywords with the higher value of $P(w_t|c_j)$ in the same class as the *PD(Q)* are chosen to expand the user's query, but the weight of those expanded terms (keywords) in the query will be downweighted by reducing the weight of original query terms.

4 Experiment

4.1 Testbed

In the experiment reports here, we examine selection and retrieval performance in distributed environments using query expansion technique with a *Naive Bayes* classifier. Our testbed was based on *the Reuters 21578 Distribution 1.0* data set that consisted of 21578 articles and 135 topic categories from the Reuters newswire [8]. The collections in this data set are indexed separately to simulate a real-world distributed IR system.

When a query is posed, the system first expands it by *Naive Bayes* learning on the training data set and then searches for it on the actual set of distributed collections. Ideally, we would like the documents in the training collections and those in the actual collections to have similar coverage of subject matters in order to expand the query properly. So, we decomposed the *Reuters 21578* data set into two subdata sets – *REUTER-TEST* used for distributed collection and *REUTER-TRAINING* which was solely for the purpose of query expansion.

General characteristics of these two subdata sets and the query sets appear in Table 1. To guarantee enough labeled training documents for *Naive Bayes* learning, we chose 96 populous class documents from 135 topic categories as the training collections of the *REUTER-TRAINING* data set. Each collection only contains relevant documents of one topic class so as to acquire

some most probable keywords about such topic class, which are calculated by *Naive Bayes* learning with these labeled training documents.

Sets of collections	*REUTER-TRAINING*	*REUTER-TEST*
Number of queries	96	96
Raw text size in megabytes	9.68	70.5
Number of documents	3294	17309
Mean words per document	176	176
Mean relevant documents per query	33	168
Number of words	5808	29568
Number of collections	96	200
Mean documents per collections	33	100

Table 1: Statistics about the sets of collections used for evaluation

4.2 Experimental Setup

In our experiments, we will consider a number of variations and evaluate the impact that these variations had on the final document retrieval results. These variations are:

- Query expansion vs without query expansion for collection selection, and for document retrieval.
- The effect of adding different number expansion concepts on retrieval effectiveness.
- The effect of varying the size of the labeled training collection on retrieval effectiveness.
- The effect of assigning different weights to the expanded concepts on retrieval effectiveness.
- The effect of increasing the number of collections selected on retrieval effectiveness

We planned experiments with these variations, and evaluated the impact that these variations had on the collection selection and on the final document retrieval result. Descriptions of the testbed, details of the selection and merging approaches, and a more detailed description of the evaluation approaches are given below.

4.3 Evaluation-Baselines for Comparison

Two baselines are referred to in the evaluation below, specifically:

(1) One is the optimal relevance-based ranking O_q for a single query q, which is used for evaluating the collection selection performance. The ranking order is produced by processing each query at each of the 200 test collections in the *REUTER-TEST* data set and then using the weight (see Equation 8 below) to rank the test collections. The algorithm for ranking test collections for a single query q is similar to the well-known $tf \cdot idf$ approach by replacing tf with df and idf with icf. It is defined as follows:

$$Weight(q \mid c_i) = df \times icf \tag{8}$$

where df is the document frequency of documents in a certain collection c_i of the *REUTER-TEST* data set. Those documents are those that belong to the same topic class as the query q; icf, inverse collection frequency, can be calculated as $\log(N / cf)$. N is the number of all collections in the *REUTER-TEST* data set, and cf is the number of collections in the *REUTER-TEST* data set which contain the same topic class documents as the query.

O_q is a ranking order where the collection with the largest weight is ranked 1, the collection with second largest weight is ranked 2, and so on.

(2) The other baseline is the retrieval performance of searching a set of distributed collections using the basic queries without query expansion. Comparison with this baseline tells us the improvement we have made by using *Naive Bayes* learning technique to expand the user's query.

5 Experimental Results

5.1 Query Expansion for Collection Selection

5.1.1 Evaluation Methodology

The mean-squared root error metric was used to compare the effectiveness of variations to the basic collection ranking algorithms. The mean-squared root error of the collection ranking for a single query is calculated as:

$$\frac{1}{|C|} \cdot \sqrt{\sum_{i \in C} (O_i - R_i)^2} \tag{9}$$

where: (1) O_i is optimal rank for collection *i,* based on the weight scorn of relevant documents it contained (see section 4.3); (2) R_i is the rank for collection *i* determined by the retrieval algorithm, which is described in the following:

$$R(q|C_i) = \sum_{t_j \in q} \sum_{d_k \in C_i} tf_{jk} \cdot \log N / df_{t_j} \tag{10}$$

where $R(q|C_i)$ is the relevant score of the query *q* in the collection c_i; tf_{jk} is the term frequency for a term T_j of the query *q* in document d_k and df_{t_j} is the number of documents in the collection c_i of *N* documents in which term T_j occurs. The collection with the largest value of $R(q|C_i)$ is ranked 1, the collection with second largest value is ranked 2, and so on; (3) *C* is the set of collections being ranked.

The mean-squared root error metric has the advantage that it is easy to understand (an optimal result is 0), and it does not require labeling a collection 'relevant' or 'not relevant' for a particular query.

5.1.2 Selection Result

Although we have argued that ranking collections is analogous to ranking documents (see Section 4.3), there are still some differences. The reason for ranking collections is *not* to find collections about a particular subject; it is to find collections containing as many documents as possible about the subject.

We first report the results of using query expansion in the collection selection stage only. As we expected, query expansion with *Naive Bayes* learning does improve collection selection. Experimental results on the REUTER-TEST data set support this, as shown by Table 2 and Figure 2. The mean-squared root errors for query expansion, averaged over 96 queries, are noticeably smaller than that for the base query.

REUTER-TEST 200-collections Testbed (96 queries)				
Mean-Squared Root Error at s collections selected	*50 Concepts*	*30 Concepts*	*10 Concepts*	*Base Query*
20 Collections	0.4847	0.4667	0.4901	0.5056
15 Collections	0.3364	0.3256	0.3347	0.3523
10 Collections	0.2763	0.2667	0.2836	0.3042
8 Collections	0.2515	0.2467	0.268	0.2923
5 Collections	0.2087	0.2016	0.224	0.2436
2 Collections	0.1423	0.1196	0.168	0.2145

Table2: The effect on mean-squared root error of varying query expansion size for selection performance on the *REUTER-TEST*

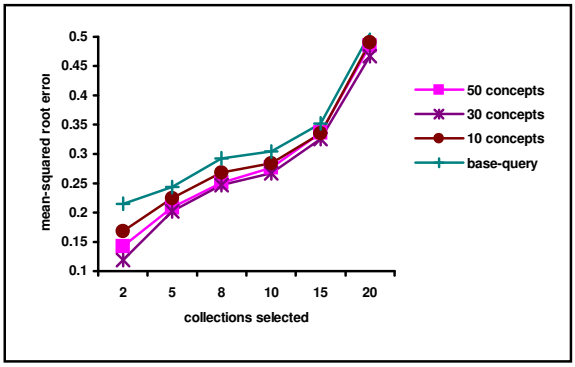

Figure 2: The effect of expansion concept size on selection performance in the REUTER-TEST

There are a number of interesting things to observe in Figure 2. Firstly, when more collections are selected for searching, mean-squared root error tends to greater. This is understandable that while selecting more collections increases the chance of selecting a relevant-rich collection. It is not guaranteed to select collections in a right order. Secondly, the greatest improvement can be seen when 30 expanded concepts are used for selection (instead of 50 concepts). For these queries, expanding more concepts does not provide a large benefit. This may be due to 30 expanded concepts which contain most relevant concepts in term to original query. Expanding more concepts may not improve performance, sometime even degrade it.

5.2 Query Expansion for Collection Selection and Retrieval

Although the improvement on collection selection is significant, we believe that the results do not reflect the power of query expansion for information retrieval. So we still expect that retrieval performance will be better than that of using the base query in both the collection selection and the retrieval stages.

Two common measures of retrieval effectiveness are *recall* and *precision* [13]. But in a realistic environment, precision at low recall is far more important, because a typical user can only afford searching a small number of documents. We only search the top 10 collections in the estimated ranking ordered by $R(q|C_i)$ for a query, and retrieve a maximum of 50 top rated documents from each collection and merge them according to their relevant weights that are calculated by the famous formula $tf \cdot idf$ [13]. In order to be able to investigate the retrieval effectiveness with query expansion, we measure the precision of the first *s* top ranked documents ranging from 10 to 100.

The goal of the experiments in this section is to confirm the effect of a number of variations concerning the benefit of query expansion on distributed information retrieval (see section 4.1).

5.2.1 Expansion Concepts

First, we compare different query expansion sizes with *Naive Bayes* learning and base query in term of retrieval effectiveness. It is interesting to see how the number of expansion concepts used affects retrieval performance. To see it more clearly, we plot the performance curve in Figure 3. Figure 3 show the effect of query expansion size on retrieval performance on REUTER-TEST compared to the retrieval baseline of base query.

Experiment results show that query expansion does improve retrieval performance if the number of expansion concepts is chosen properly. Reducing the number of concepts from 50 to 30 does not apparently affect retrieval effectiveness. In fact, using 30 concepts is even slightly better than using 50 concepts. But when only 10 concepts are used per query, retrieval performance suffers, by 9.82% on average. One possible problem is that query expansion with only 10 concepts cannot provide some so-called topic words which by themselves

are very strong indicators of relevance. So those non-relevant expansion concepts hurt retrieval effectiveness.

Analysis of the results also reveals that for more than 30 expansion concepts, retrieval is improved at all documents cut-offs. Improvement at higher cut-offs (from 70 to 100) is around 7%, which is more noticeable than at lower cut-offs.

5.2.2 Selection Collections

We are also interested in the impact of selecting more or fewer collections to search on retrieval performance. In general case, selecting more collections increases the chances of selecting a relevant-rich collection with the most (or even any) relevant documents. It is surprising that the greatest improvement can be seen when 10 collections are selected (instead of 15 or 20). This can be seen in Figure 4. This may be explained by a phenomenon – there are queries for which many relevant documents can be found in the top 10 collections. For these queries, searching a larger number of collections does not provide a large benefit.

Searching additional collections tends to improve retrieval performance, but there are limits to that trend. In fact, beyond a certain point, searching additional collections may degrade performance.

5.2.3 Training Collections

Large training sets are required to provide a useful classification and to get accurate expansion concepts for *Naive Bayes* learning. Since it is tedious and expensive to create these sets of labeled data, we naturally consider the impact of using smaller training collections. So instead of using the full REUTER-TRAIN collections for query expansion, we vary the amount of labeled training data by 75%, 50% and 25% of the REUTER-TRAIN to get expansion concepts.

Figure 5 show retrieval results. It understands that the full TRAIN has the best performance at a large labeled data. There is a rapid decrease in performance as 25% of the labeled data in TRAIN is used. Comparing with using full TRAIN, there is only a small degradation, especially at higher cut-offs (about 2.4%) when 75% and 50% of TRAIN are used for query expansion. It suggests that it is possible to cut the size of the training collection without significantly affecting retrieval effectiveness. However, currently we do not know how to automatically determine the optimal size of the training set. We need to do the further investigations to solve the problem in our future research.

5.2.4 Weight of Expansion Concepts

The high baseline of the REUTER-TEST data set (46.3% average precision) suggests that the original queries are of very good quality and we should give them more emphasis. So, we add a parameter that varies the relative contribution of expansion concepts on retrieval performance. Figure 6 show that downweighting the expansion concepts does improve performance. Experiments are conducted with weight values ranging from 0.2 to 1. The results indicate that when we downweight the expansion concepts by 80% by reducing the weight of query from 1 to 0.2, the retrieval performance is slightly better than other weight values. It suggests that although expansion concepts help to improve retrieval effectiveness, we should pay more attention on the base query in case that improper expansion concepts hurt retrieval performance.

6 Conclusions and Future Work

The problem of word mismatch has become an important issue for searching distributed collections, as more and more information sources are available online and heterogeneously distributed over the Internet.

This paper describes an efficient technique to address this problem based on query expansion with *Naive Bayes*. Such technique can make a system to automatically add other topic terms related to the same concepts in a user query to effectively rank collections and search the subset efficiently. The effectiveness of the technique is demonstrated in experiments with IISS system and the *Reuters 21578* data set.

The experimental results on the *Reuters 21578* data set are extremely encouraging. They suggest that it is possible to improve the effectiveness on both selection and retrieval stages in distributed searching environments by using query expansion with a *Naive Bayes classifier*.

However, there are a number of areas in which we will continue our work. Firstly, we plan to use even larger collections such as the 20 Gigabytes TREC VLC (Very Large Corpus) collection to test our techniques. Secondly, we try to find a "versatile" training collection for query expansion. Such a collection should have a wide coverage of subject matters so that most queries can be properly expanded.

7 Acknowledgement

This research was supported by a large grant from the Australian Research Council under contract DP0211282.

8 References

[1] J. P. Callan, W. B. Croft and S. M. Harding, "The INQUIRY Retrieval System," Proc. the third International Conference on Database and Expert System Application, Valencia, Spain, pp. 78-83, 1992.

[2] J. P. Callan, Z. Lu, and W. B. Croft, "Searching Distributed Collections with Inference Networks," Proc. of SIGIR'95, Seattle, pp. 21-29, 1995.
[3] J. Callan, A, L, Powell, "The effects of Query-based Sampling," Technique Report LMU-LTI-00162. Carnegie Mellon University, 2000.
[4] N. Fuhr, "A Decision-Theoretic Approach to Database Selection in Networked IR," ACM Transactions on Information Systems, vol.17, issue 3, pp. 229-249, 1999.
[5] J. C. French, A. L. PoweU, J. Callan, C. L. Viles, T. Emmitt, K. J. Prey, and Y. Mou, "Comparing the Performance of Database Selection Algorithms," Proc. of SIGIR'99, Berkeley, pp. 238-245, 1999.
[6] J. C. French, A. L. Powell, C. L. Viles, T. Emmitt, and K. J. Prey, "Evaluating Database Selection Techniques: A Testbed and Experiment," Proc. of SIGIR'98, Melbourne Australia, pp. 121-129, 1998.
[7] L. Gravano, H. Garcia-Molina, and A. Tomasic, "The Effectiveness of GLOSS for the Text Database Discovery Problem," Proc. SIGMOD94, Philadelphia, pp. 126 – 137, 1994.
[8] D. D. Lewis, "Reuter-21578 Text Categorization Test Collection Distribution 10," Available at http://www.research.att.com/~lewis.
[9] D. D. Lewis, "*Naive* Bayes at Forty: The Independence Assumption in Information Retrieval," Proc. EDML-98, Chemnitz, Germany, 1998.
[10] Andrew McCallum and Kamal Nigam, "A Comparison of Event Models for Naive Bayes Text Classification" Proc. AAAI-98 Workshop on Learning for Text Categorization, 1998.
[11] M. Porter, "An Algorithm for Suffix Stripping," Program vol.14, 1980.
[12] M. Schwartz, A. Emtage, B. Kahle, and B. Neumann, "A comparison of Internet Resource Discovery Approaches," Computer Systems, vol. 5, issue 4, pp. 461-493, 1992.
[13] G. Salton and M. Mcgill, Introduction of Modern Information Retrieval, McGrag-Hill, 1983.
[14] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird, "The Collection Fusion Problem," Proc. The Third Text Retrieval, Maryland, pp. 95-104, 1995.
[15] J. Xu and W. B. Croft, "Cluster-based Language Models for Distributed Retrieval," Proc. of SIGIR'99, Berkeley, pp. 254-261, 1999.
[16] B. Yuwono and D. L. Lee, "Server Ranking for Distributed Text Retrieval Systems on Internet," Proc. the Conference on Database Systems for Advanced Applications, New Orleans, pp. 41- 49, 1997.
[17] Hui Yang, Minjie Zhang and Xiaohua Yang, "IISS: A Framework for Intelligent Information Source Selection on the Web," Proc. International Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand, pp. 209-215, 2001.

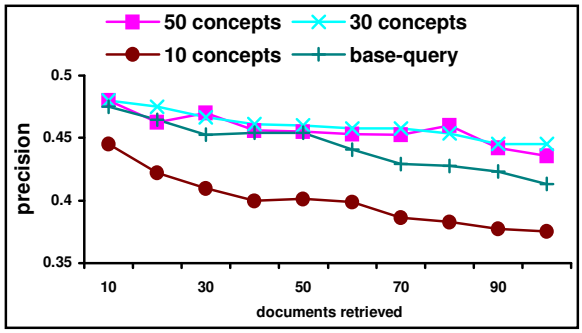

Figure 3: The effect of query expansion size on retrieval performance in the REUTER-TEST

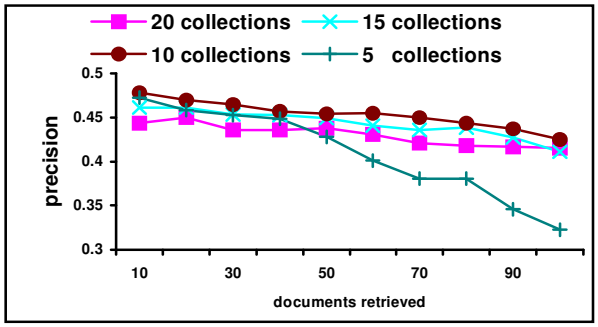

Figure 4: The effect of selection collection size on retrieval performance in the REUTER-TEST

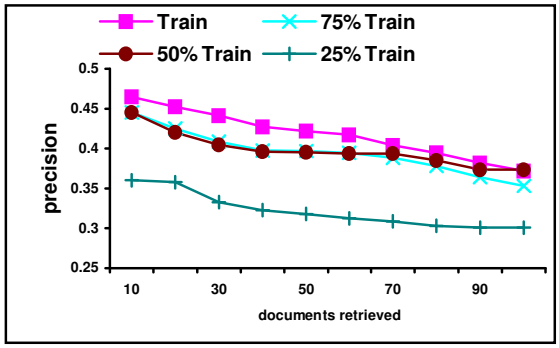

Figure 5: The effect of the training collection size on retrieval performance in the REUTER-TEST

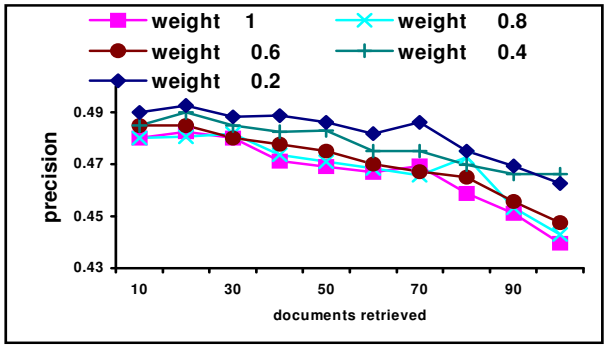

Figure 6: The effect of the different weight of expansion concepts on retrieval performance in the REUTER-TEST