Probabilistic Query Expansion Using Query Logs

Hang Cui[1*], Ji-Rong Wen[2], Jian-Yun Nie[3*], Wei-Ying Ma[2]
[1]Tianjin University, Tianjin, P.R.China, Email: hang_cui@hotmail.com
[2]Microsoft Research Asia, Beijing, P.R.China, Email: {jrwen, wyma}@microsoft.com
[3]Université de Montréal, Email: nie@iro.umontreal.ca

ABSTRACT
Query expansion has long been suggested as an effective way to resolve the short query and word mismatching problems. A number of query expansion methods have been proposed in traditional information retrieval. However, these previous methods do not take into account the specific characteristics of web searching; in particular, of the availability of large amount of user interaction information recorded in the web query logs. In this study, we propose a new method for query expansion based on query logs. The central idea is to extract probabilistic correlations between query terms and document terms by analyzing query logs. These correlations are then used to select high-quality expansion terms for new queries. The experimental results show that our log-based probabilistic query expansion method can greatly improve the search performance and has several advantages over other existing methods.

Categories and Subject Descriptors
H.2.8 [Database Management]: Database Applications – Data mining; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval.

General Terms
Algorithms, Performance, Experimentation, Human Factors

Keywords
Query expansion, log mining, probabilistic model, information retrieval, search engine

1. INTRODUCTION
With the explosive growth of information on the World Wide Web, there is an acute need for search engine technology to help users exploit such an extremely valuable resource. Despite the fact that keywords are not always good descriptors of contents, most existing search engines still rely solely on the keywords contained in the queries to search and rank relevant documents. This is one of the key reasons that affect the precision of the search engines. In many cases, the answers returned by search engines are not relevant to the user information need, although they do contain the same keywords as the query. The web is not a well-organized information source where innumerous "authors" created and are creating their websites independently. Therefore, the "vocabularies" of the authors vary greatly. On the other hand, users usually tend not to use the same terms appearing in the documents as search terms. This raises a fundamental problem of term mismatch in information retrieval. Moreover, most words in natural language have inherent ambiguity. These reasons make it a rather difficult task for the web users to formulate queries with appropriate words.

It is also generally observed that web users typically submit very short queries to search engines and the average length of web queries is less than two words [16]. Short queries usually lack sufficient words to cover useful search terms and thus negatively affect the performance of web search in terms of both precision and recall.

To overcome the above problems, researchers have focused on using query expansion techniques to help users formulate a better query. Query expansion involves adding new words and phrases to the existing search terms to generate an expanded query. However, previous query expansion methods have been limited in extracting expansion terms from a subset of documents, but have not exploited the accumulated information on user interactions. We believe that this latter is extremely useful for adapting a search engine to the users. In particular, we will be able to find out what queries have been used to retrieve what documents, and from that, to extract strong relationships between query terms and document terms and to use them in query expansion.

In this paper, we suggest a new query expansion method based on the analysis of user logs. By exploiting correlations among terms in documents and user queries mined from user logs, our query expansion method can achieve significant improvements in retrieval effectiveness compared to current query expansion techniques.

2. RELATED WORK
The existing state-of-the-art query expansion approaches can be classified mainly into two classes – global analysis and local analysis.

Global analysis is one of the first techniques to produce consistent and effective improvements through query expansion. One of the earliest global analysis techniques is term clustering [15], which groups document terms into clusters based on their co-occurrences. Queries are expanded by the terms in the same cluster. Other well-known global techniques include Latent Semantic Indexing [4], similarity thesauri [11], and PhraseFinder [8]. Global analysis requires corpus-wide statistics such as statistics of co-occurrences of pairs of terms, which results in a similarity matrix among terms. To expand a query, terms which are the most similar to the query terms are identified and added. The global analysis techniques are relatively robust; but corpus-wide statistical analysis consumes a considerable amount of

WWW 2002, May 7 - 11, 2002, Honolulu, Hawaii, USA.
ACM 1-58113-449-5/02/0005.

* This work was performed while the authors were visiting Microsoft Research Asia

computing resources. Moreover, since it only focuses on the document side and does not take into account the query side, global analysis cannot address the term mismatch problem well.

Different from global analysis, local analysis uses only some initially retrieved documents for further query expansion. The idea of local analysis can be traced back at least to a 1977 paper [1]. A well-known local analysis technique is relevance feedback [12, 14], which modifies a query based on user's relevance judgments of the retrieved documents. Typically, expansion terms are extracted from the relevant documents. Relevance feedback can achieve very good performance if the users provide sufficient and correct relevance judgments. Unfortunately, in a real search context, users usually are reluctant to provide such relevance feedback information. Therefore, relevance feedback is seldom used by the commercial search engines.

To overcome the difficulty due to the lack of sufficient relevance judgments, pseudo-relevance feedback (also known as blind feedback) is commonly used. Local feedback mimics relevance feedback by assuming the top-ranked documents to be relevant [3, 13]. Expansion terms are extracted from the top-ranked documents to formulate a new query for a second cycle retrieval.

In recent years, many improvements have been obtained on the basis of local feedback, including re-ranking the retrieved documents using automatically constructed fuzzy Boolean filters [10], clustering the top-ranked documents and removing the singleton clusters [9], clustering the retrieved documents and using the terms that best match the original query for expansion [2]. In addition, recent TREC results show that local feedback approaches are effective and, in some cases, outperform global analysis techniques [17]. Nevertheless, this method has an obvious drawback: if a large fraction of the top-ranked documents is actually irrelevant, then the words added to the query (drawn from these documents) are likely to be unrelated to the topic and as a result, the quality of the retrieval using the expanded query is likely to be worse. Thus the effects of pseudo-feedback strongly depend on the quality of the initial retrieval.

Recently, Xu and Croft [18] proposed a local context analysis method, which combines both local analysis and global analysis. First, noun groups are used as concepts, which are selected according to their co-occurrences with the query terms. Then concepts are chosen from the top-ranked documents, similarly to local feedback. Since expansion terms used here are not based on frequencies in the top-ranked documents but rather on co-occurrences with terms in the query, the local context analysis method can overcome the difficulty of local analysis to some extent. However, local context analysis is based on the hypothesis that a frequent term from the top-ranked relevant documents will tend to co-occur with all query terms within the top-ranked documents. This is a reasonable hypothesis, but not always true.

3. PRINCIPLE OF USING QUERY LOGS

We observe that many search engines have accumulated a large amount of query logs, from which we can know what the query is, and what the documents the user has selected to read. These query logs provide valuable indications to understand the kinds of documents the users intend to retrieve by formulating a query with a set of particular terms.

In this study, we put forward a query expansion method based on an exploitation of query logs. From the query logs we can extract many query sessions, which are defined as follows:

session := <query text> [clicked document]*

Each session contains one query and a set of documents which the user clicked on (which we will call clicked documents). The central idea of our method is that if a set of documents is often selected for the same queries, then the terms in these documents are strongly related to the terms of the queries. Thus some probabilistic correlations between query terms and document terms can be established based on the query logs. These probabilistic correlations can be used for selecting high-quality expansion terms from documents for new queries.

One important assumption behind this method is that the clicked documents are relevant to the query. This assumption may appear too strong. However, although the clicking information is not as accurate as explicit relevance judgment in traditional IR, the user's choice does suggest a certain degree of relevance. In fact, users usually do not make the choice randomly. In addition, we benefit from a large quantity of query logs. Even if some of the document clicks are erroneous, we can expect that most users do click on documents that are, or seem to be, relevant. Our previous work on using query logs to cluster similar queries also strongly supports this assumption [16]. Therefore, query logs can be taken as a very valuable resource containing abundant relevance feedback data. Thus we can overcome the problem of lacking sufficient relevance judgments in traditional relevance feedback technique. In comparison with pseudo-relevance feedback, our method has an obvious advantage: not only are the clicked documents part of the top-ranked documents, but also there is a further selection by the user. So document clicks are more reliable indications than those used in pseudo relevance feedback.

The log-based query expansion method has three other important properties. First, since the term correlations can be pre-computed offline, the initial retrieval phase is not needed anymore. Second, since query logs contain query sessions from different users, the term correlations can reflect the preference of most users. For example, if the majority of users use "windows" to search for information about Microsoft Windows product, the term "windows" will have much stronger correlations with the terms such as "Microsoft", "OS" and "software" rather than with the terms such as "decorate", "door" and "house". Thus the expanded query will result in a higher ranking for the documents about Microsoft Windows. The similar idea has been used in several existing search engines, such as Direct Hit [5]. Our query expansion approach can produce the same results. Third, the term correlations may evolve along with the accumulation of user logs. The query expansion process can reflect updated user's interests at a specific time.

4. PROBABILISTIC QUERY EXPANSION BASED ON QUERY LOGS

In this section, we describe in more detail the log-based query expansion method. First, we will test the assumption that the terms used in queries and in documents are truly very different. This assumption has often been made, but never tested by a quantitative measurement. Our test will show that there is indeed a large difference between the query terms and document terms.

Therefore some mechanisms are needed to bridge the gap, that is, to build up the relationships between query terms and document terms. We then discuss our extraction of term correlations by mining the query logs. Finally, we will show how these term correlations are used in query expansion.

4.1 Gap between the Query Space and the Document Space

Let us define all the term usages in the documents as forming a document space, and those in the queries as a query space. Inconsistency between term usages in queries and documents is a well-known problem in information retrieval. This is one of the very facts that motivate the use of query expansion. This problem was first observed by Furnas [6] in a more general context. It is even worse when the query is very short as is the case on the web.

Until now, no one has precisely measured how different the two spaces are. This is difficult without user query logs. In order to arrive at a quantitative measure, we make use of two-month query logs (about 22 GB) from the Encarta search engine (http://encarta.msn.com), as well as the 41,942 documents in the Encarta website. From these logs we extracted 4,839,704 user query sessions. Below is an excerpt of the query sessions.

Queries	IDs of clicked documents
Trinidad and Tobago	761561556 761559363
Amish pacifism	761586809
Electric lights	761579230
Marion Jones	761562123
Ben Johnson	761562123
Spoils System	761551930
Indian removal act	761553925
Pecan tree pictures	761572753
New Mexico	761572098 761572098

Each document can be represented as a document vector $\{W_1^{(d)}, W_2^{(d)} \ldots W_n^{(d)}\}$ in the document space, where $W_i^{(d)}$ is the weight of the i^{th} term in a document and is defined by the traditional TF-IDF measure:

$$W_i^{(d)} = \frac{\ln(1 + tf_i^{(d)}) \times idf_i^{(d)}}{\sqrt{\sum \ln^2(1 + tf_i^{(d)}) \times \sum (idf_i^{(d)})^2}} \quad (1)$$

$$idf_i^{(d)} = \ln \frac{N}{n_i} \quad (2)$$

where $tf_i^{(d)}$ is the frequency of the i^{th} term in the document D, N is the total number of documents in the collection, and n_i the number of documents containing the i^{th} term. For each document, we can construct a corresponding virtual document in the query space by collecting all the queries for which the document has been clicked on. A virtual document is represented as a query vector $\{W_1^{(q)}, W_2^{(q)} \ldots\ldots W_n^{(q)}\}$ where $W_i^{(q)}$ is the weight of the i^{th} term in the virtual document and also is defined by the TF-IDF measure.

To compare the query space and document space, we only need to measure the similarity between the document vector and its corresponding query vector. Specially, the similarity of each pair of vectors can be measured by using the following Cosine similarity:

$$Similarity = \frac{\sum_{i=1}^{n} W_i^{(q)} W_i^{(d)}}{\sqrt{\sum_{i=1}^{n} W_i^{2(q)}} \sqrt{\sum_{i=1}^{n} W_i^{2(d)}}} \quad (3)$$

We noticed that many terms in the document space are never or seldom used in the users' queries. Thus many terms in the document vector appear with very small or otherwise zero weights in its corresponding query vector. This artifact will dramatically decrease the similarity between the two vectors if they are used in the measurement. To obtain a fairer measure, we only use the most important words in the document vectors for the similarity calculation, where is the number of terms in the virtual document.

Figure 1 illustrates the final results of similarity values on the whole document collection. This figure shows that, in most cases, the similarity values of term usages between user queries and documents are between 0.1 and 0.4. Only very few documents have similarity values above 0.8. The average similarity value across the whole document collection is 0.28, which means the average internal angle between the query vector and the document vector is 73.68 degree. This result confirms that there is indeed a large gap between the query space and the document space. It is then important to find ways to narrow the gap, or to bridge the two spaces in order to improve the retrieval effectiveness.

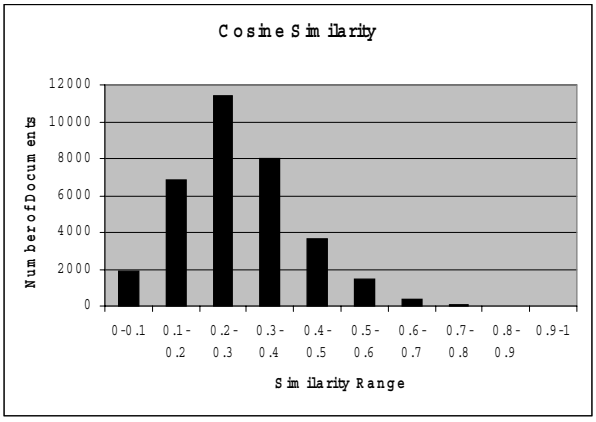

Figure 1. Similarity between the query terms and document terms

4.2 Correlations between Query Terms and Document Terms

Query sessions in the query logs provide a possible way to bridge the gap between the query space and the document space. Figure 2 shows how correlations between the query terms and document terms can be established through the query sessions. In general, we assume that the terms in a query are correlated to the terms in

the documents that the user clicked on. If there is at least one path between one query term and one document term, a link is created between them. By analyzing a large numbers of such links, we can obtain a probabilistic measure for the correlations between the terms in these two spaces (Figure 3).

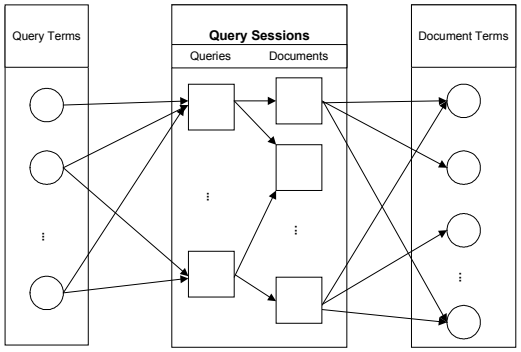

Figure 2. Query sessions, query terms and document terms

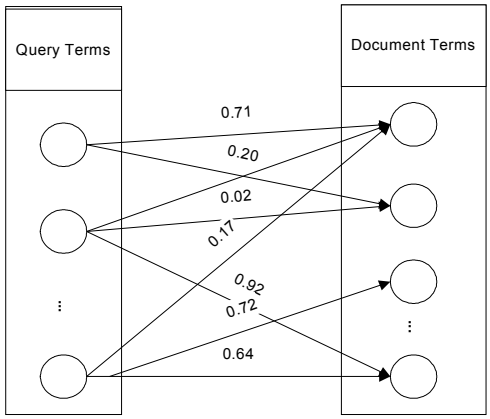

Figure 3. Probabilistic correlations between query terms and document terms

Let us now discuss how to determine the degrees of correlations between terms. We define degrees as the conditional probabilities between terms, i.e. $P(w_j^{(d)} \mid w_i^{(q)})$. Let $w_j^{(d)}$ and $w_i^{(q)}$ be an arbitrary document term and a query term, respectively. The probability $P(w_j^{(d)} \mid w_i^{(q)})$ can be determined as follows (where S is a set of clicked documents for queries containing the query term $w_i^{(q)}$):

$$P(w_j^{(d)} \mid w_i^{(q)}) = \frac{P(w_j^{(d)}, w_i^{(q)})}{P(w_i^{(q)})}$$

$$= \frac{\sum_{\forall D_k \in S} P(w_j^{(d)}, w_i^{(q)} \mid D_k) \times P(D_k)}{P(w_i^{(q)})}$$

$$= \frac{\sum_{\forall D_k \in S} P(w_j^{(d)}, w_i^{(q)}, D_k)}{P(w_i^{(q)})}$$

$$= \frac{\sum_{\forall D_k \in S} P(w_j^{(d)} \mid w_i^{(q)}, D_k) \times P(w_i^{(q)}, D_k)}{P(w_i^{(q)})}$$

We can assume that $P(w_j^{(d)} \mid w_i^{(q)}, D_k) = P(w_j^{(d)} \mid D_k)$ because the document D_k separates the query term $w_i^{(q)}$ from the document term $w_j^{(d)}$.

$$= \frac{\sum_{\forall D_k \in S} P(w_j^{(d)} \mid D_k) \times P(D_k \mid w_i^{(q)}) \times P(w_i^{(q)})}{P(w_i^{(q)})} \quad (4)$$

$$= \sum_{\forall D_k \in S} P(w_j^{(d)} \mid D_k) \times P(D_k \mid w_i^{(q)})$$

$P(D_k \mid w_i^{(q)})$ is the conditional probability of the document D_k being clicked in case that $w_i^{(q)}$ appears in the user query. $P(w_j^{(d)} \mid D_k)$ is the conditional probability of occurrence of $w_j^{(d)}$ if the document D_k is selected. $P(D_k \mid w_i^{(q)})$ and $P(w_j^{(d)} \mid D_k)$ can be estimated respectively from the query logs and from the frequency of occurrences of terms in documents as follows:

$$P(D_k \mid w_i^{(q)}) = \frac{f_{ik}^{(q)}(w_i^{(q)}, D_k)}{f^{(q)}(w_i^{(q)})} \quad (5)$$

$$P(w_j^{(d)} \mid D_k) = \frac{W_{jk}^{(d)}}{\max_{\forall t \in D_k}(W_{tk}^{(d)})} \quad (6)$$

where

$f_{ik}^{(q)}(w_i^{(q)}, D_k)$ is the number of the query sessions in which the query word $w_i^{(q)}$ and the document D_k appear together.

$f^{(q)}(w_i^{(q)})$ is the number of the query sessions that contain the term $w_i^{(q)}$.

$P(w_j^{(d)} \mid D_k)$ is the normalized weight of the term $w_j^{(d)}$ in the document D_k, which is divided by the maximum value of term weights in the document D_k.

By combining the formulas (4), (5) and (6), we obtain the following formula to calculate $P(w_j^{(d)} \mid w_i^{(q)})$.

$$P(w_j^{(d)} \mid w_i^{(q)}) = \sum_{\forall D_k \in S} (P(w_j^{(d)} \mid D_k) \times \frac{f_{ik}^{(q)}(w_i^{(q)}, D_k)}{f^{(q)}(w_i^{(q)})}) \quad (7)$$

4.3 Query Expansion Based on Term Correlations

Our query expansion method is based on the probabilistic term correlations described above. When a new query is submitted, first, the terms in the query (after removing stop words) are extracted. Then for every query term, all correlated document terms are selected based on the conditional probability obtained by the formula (7). By combining the probabilities of all query terms, we can calculate the following cohesion weight of a document term for the new query Q:

$$CoWeight_Q(w_j^{(d)}) = \ln(\prod_{w_t^{(q)} \in Q} (P(w_j^{(d)} \mid w_t^{(q)}) + 1)) \quad (8)$$

Thus, for every query, we get a list of weighted candidate expansion terms. The top-ranked terms can be selected as expansion terms.

5. EVALUATION

In this section, we report the experimental results on the performance of the log-based probabilistic query expansion method.

5.1 Data

We used the same two-month query logs from the Encarta website as described in Section 4.1, which contains 4,839,704 user query sessions. The documents collection is made up of 41,942 Encarta documents with various topics. The lengths of the documents also vary greatly, from dozens of words to several thousand words.

A total of 30 queries are used to conduct the experiments. Some queries are extracted randomly from the query logs. Some others come from the TREC query set. Yet another subset of queries are added manually by ourselves. The queries in our experiments are very close to those employed by the real web users and the average length of all queries is 2.1 words. Figure 4 lists the 30 queries used in the experiments.

1 Java computer	2 nuclear submarine	
3 Apple computer	4 Windows	5 fossil fuel
6 cellular phone	7 search engine	
8 Six Day War	9 space shuttle	
10 economic impact of recycling tires		
11 China Mao Ze Dong	12 atomic bomb	
13 Manhattan project	14 Sun Microsystems	
15 Cuba missile crisis	16 motion pictures	
17 Steve Jobs	18 pyramids	19 what is Daoism
20 Chinese music	21 genome project	
22 Apollo program	23 desert storm	
24 table of elements	25 Toronto film awards	
26 Chevrolet truck	27 DNA testing	
28 Michael Jordan	29 Ford	30 ISDN

Figure 4. The experimental query set

Relevant documents are judged according to the human assessors' manual selections and standard relevant document sets are prepared for all of the 30 queries.

5.2 Word and Phrase Thesaurus

Encarta has well-organized manual indexes in addition to automatically extracted index terms. In order to test our techniques in a general context, we did not use manual indexes or the existing Encarta searching engine for evaluation. Instead, we implement a vector space model as the baseline method in our experiments.

We do not use traditional methods to extract phrases from documents because we are more interested in the phrases in the query space. Therefore, we extract all N-grams from the query logs with occurrences higher than 5 and treat the N-grams as candidate phrases. Then we relocate these candidate phrases in the document corpus and filter out those not appearing in the documents. In the end we get a thesaurus containing over 13,000 phrases.

We notice that the occurrences of phrases are far less than those of words. This creates an unbalance between the weights we assigned to word correlations and to phrase correlations. In order to create a better balance, the probability associated with a phrase correlation is multiplied by a factor S . The formula used to measure phrase correlations is modified from Formula (7) to the following one:

$$P(w_j^{(d)} \mid w_i^{(q)}) = \sum_{\forall D_k \in S} (P(w_j^{(d)} \mid D_k) \times \frac{S \times f_{ik}^{(q)}(w_i^{(q)}, D_k)}{f^{(q)}(w_i^{(q)})}) \quad (9)$$

This method is simple. One can also use idf as an alternative, since in general the occurrences of phrases are less than that of words, thus phrases have higher idf values.

5.3 Quality of Expansion Terms

We examined the top 50 expansion terms for all the 30 queries to check the relevance of the expansion terms. The Table 1 shows the number of relevant expansion terms suggested by the local context analysis (LC Analysis) and our log-based method (Log Based). As we can see, our method is 32.03% better.

Table 1. Comparison on relevant percentage of expansion terms (Top 50 terms)

	LC Analysis (base)	Log Based	Improvement (%)
Relevant Terms (%)	23.27	30.73	+32.03

Figure 5 illustrates the top 50 expansion terms for the query "Steve Jobs" by our method. Some very good terms, such as "personal computer", "Apple Computer", "CEO" , "Macintosh", even "graphical user interface", "Microsoft" can be obtained by our techniques.

1.Apple	2. ***personal computer***	3. ***Computers***
4***. personal computers***	5. ***Apple Computer***	
6. ***operating system***	7. ***Newton***	
8. ***graphical user interface***	9. graphical user	
10. ***Software***	11. user interface	
12. programming language	13. programming languages	
14. ***computer***	15. wozniak	
16. CPU	17. operating systems	
18. mainframe computer	19. personal	
20. principia	21. jobs	
22. ***CEO***	23. company	
24. computer systems	25. high-level	
26. assembly language	27. machine language	
28. computer system	29. ***Gates***	
30. analog	31. circuit board	
32. vice president	33. opticks	
34. analytical engine	35. ***Microsoft***	
36. jacquard	37. output devices	
38. Halley	39. woolsthorpe	
40. output device	41. Calculus	
42. input devices	43. ***Lisa***	
44. Pixar	45. first computer	
46. Paul Allen	47. white light	
48. ***Macintosh***	49. slide rule	50. markkula

Figure 5. Expansion terms for "Steve Jobs"

5.4 Retrieval Effectiveness

Now we compare the retrieval performance of the log-based query expansion with the baseline (without query expansion) and the local context analysis method. Interpolated 11-point average precision is employed as the main metric of retrieval performance. Statistical paired t-test [7] is also used to determine the significance of differences.

For the local context analysis, the default is to use 30 expansion terms, which include words and phrases, from 100 top-ranked documents for query expansion. The smoothing factor δ in local context analysis is set to 0.1 here, as proposed in [18].

For the log-based query expansion, we use 40 expansion terms. The expansion terms are extracted from top 100 relevant documents according to the query logs. Phrases that appear in the query logs are assigned a parameter *S*, which is 10 in our experiments.

The retrieval results are shown in Table 2 and Figure 6.

Table 2. A comparison of retrieval performance of Baseline, query expansion base on local context analysis (LC Exp), and log-based query expansion (On Log Exp). All experiments are done with phrases included.

Recall	Baseline	LC Exp	On Log Exp
10	40.67	40.33(-0.82)	62.00(+52.46)
20	26.83	33.33(+24.22)	44.67(+66.46)
30	21.56	27.00(+25.26)	37.00(+71.65)
40	17.75	23.08(+30.05)	31.50(+77.46)
50	15.07	20.40(+35.40)	27.67(+83.63)
60	13.00	17.89(+37.61)	24.56(+88.89)
70	11.43	16.29(+42.50)	22.24(+94.58)
80	10.17	15.08(+48.36)	20.42(+100.82)
90	9.44	13.96(+47.84)	18.89(+100.00)
100	8.70	13.07(+50.19)	17.37(+99.62)
Average	17.46	22.04(+26.24)	30.63(+75.42)

We see that our log-based query expansion performs well on the experiments. It brings an improvement of 75.42% improvement in average precision (p-value=0.0000039585) over the baseline method, while the local context analysis achieves a 26.24% improvement in average precision (p-value=0.018648254) over the baseline. The p-values of both tests show that both improvements are statistical significant. The p-values also indicate that the improvement gained by our method is statistically more significant than the local context analysis. The average precision 17.46% of baseline is lower than that obtained by TREC experiments, which may be attributed to the fact that queries in our experiments are much shorter than those in the TREC. This is closer to the real scenario on the Internet. In addition, local context analysis also produces a large (26.24%) improvement compared to the baseline. That is a noticeable improvement, even slightly higher than the result reported in [18], which obtains 23.3% improvement on TREC3 and 23.5% improvement on TREC4. That indicates that query expansion is extremely important for short queries.

Log-based query expansion also provides an average improvement of 38.95% compared to local context analysis, which is also statistically significant (p-value= 0.000493316). Generally, log-based query expansion selects expansion terms from a relatively narrower but more concentrated area. In contrast, local context analysis searches expansion terms in the top-ranked retrieved documents and is more likely to add some irrelevant terms into the original query, thus introducing undesirable side effects on retrieval performance.

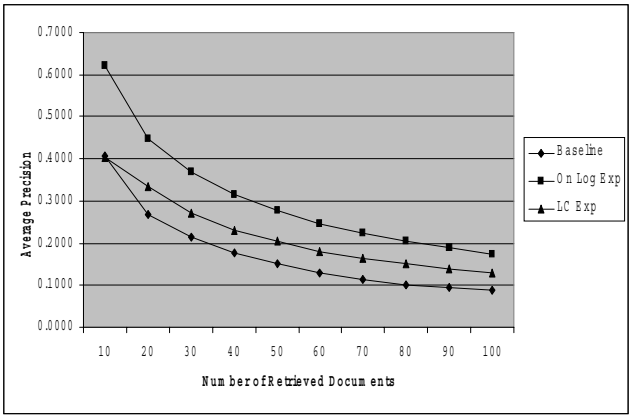

Figure 6. Average precision for Baseline, LC Exp and On Log Exp (with phrases)

5.5 Impact of Phrases

In [18], it is indicated that using noun phrases has little impact on retrieval performance. According to their experiments, the performance decreases by only 0.2% on TREC4 without using phrases. For the TREC queries, there is enough information to tradeoff the advantages of phrases since the queries are relatively long (7.5 words on average per query in TREC4). But for short queries, phrases are of crucial importance because they are more accurate representations of information and requirements. Without phrases, separate words in the query may lead to poor results. For example, given the query "search engine", if it is represented as "search" and "engine", few of the retrieved documents will be related to search engine, and most of them pertain to mechanical engines. Our experiments show that the performance can be improved greatly when phrases are introduced into the query expansion and retrieval phases. On average, an 11.37% improvement can be obtained (see Table 3).

Table 3. Comparison of average precision (%) obtained by query expansion with phrases (Phrase) and without phrases (No Phrase)

Recall	No Phrase	Phrase	Improvement (%)
10	53.00	62.00	16.98
20	40.67	44.67	9.84
30	32.56	37.00	13.65
40	28.67	31.50	9.88
50	25.47	27.67	8.64
60	22.78	24.56	7.81
70	20.43	22.24	8.86
80	18.63	20.42	9.62
90	17.04	18.89	10.87
100	15.80	17.37	9.92
Average	27.50	30.63	11.37

5.6 Impact of Number of Expansion Terms

In general, the number of expansion terms should be within a reasonable range in order to produce the best performance. Too many expansion terms not only consume more time in retrieval process, but also have side effects on retrieval performance.

We examine performance by using 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 expansion terms for retrieval. Table 4 and Figure 7 show the impact on average precision by the number of expansion terms. The best performances are obtained within the range of 40 and 60 terms. The performance drops when the number of expansion terms is larger than 70, which indicates that the terms beyond 70 are less relevant to the original query.

Table 4. Comparison of performance using various number of expansion terms

Number of Expansion Terms	Precision
10	0.271
20	0.294
30	0.303
40	0.306
50	0.306
60	0.308
70	0.304
80	0.304
90	0.302
100	0.296

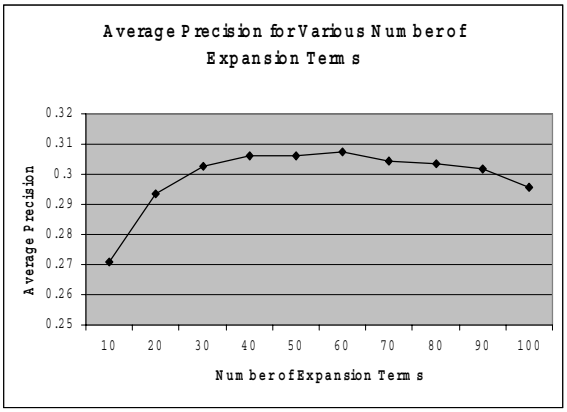

Figure 7. Impact of various number of expansion terms

6. CONCLUSION

The proliferation of the World Wide Web prompts the wide application of search engines. However, short queries and inconsistency between user query terms and document terms

strongly affect the performance of existing search engines. Many automatic query expansion techniques have been proposed. They can solve the short query and term mismatch problem to some extent. However, they fail to take advantage of the query logs available in various websites, and use them as a means for query expansion.

In this article, we presented a novel method for automatic query expansion based on query logs. This method aims to establish correlations between query terms and document terms by exploiting the query logs. We have shown that this is an effective way to narrow the gap between the query space and the document space. For new queries, high-quality expansion terms can be selected from the document space on the basis of these probabilistic correlations. We tested this method on a data set that is similar to the real web environment. A series of experiments showed that the log-based method can achieve substantial performance improvements, not only over the baseline method without expansion, but also with respect to the local context analysis, which has been proven one of the most effective query expansion methods in the past.

Query expansion using query logs is only one application of web log mining. Other useful knowledge can be obtained through analyzing the users' behaviors recorded in the logs. We believe this is a very promising research direction.

7. ACKNOWLEDGEMENTS

The authors are grateful to Zheng Zhang for his valuable comments and suggestions.

8. REFERENCES

[1] Attar, R. and Fraenkel, A.S. 1977. Local feedback in full-text retrieval systems. J. ACM 24, 3 (July), 397-417.

[2] Buckley, C., Mitra, M., Walz, J. and Cardie, C. 1998. Using clustering and superconcepts within SMART. Proceedings of the 6th text retrieval conference (TREC-6), E. Voorhees, Ed. 107-124. NIST Special Publication 500-240.

[3] Buckley, C., Salton, G., Allan, J., and Singhal, A., 1995, Automatic query expansion using SMART, TREC 3. Overview of the Third Text REtrieval Conference (TREC-3), pages 69--80. NIST, November 1994. http://trec.nist.gov/.

[4] Deerwester, S., Dumai, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. 1990. Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41,6, Pages 391-407.

[5] Direct Hit website. http://www.directhit.com/.

[6] Furnas, G.W., Landauer, T.K., Gomez, L.M. and Dumais, S.T. 1987. The vocabulary problem in human-system communication. Commun. ACM 30, 11 (Nov. 1987), Pages 964-971.

[7] Hull, D., 1993, Using statistical testing in the evaluation of retrieval experiments. In Proceedings of the ACM SIGIR, pages 329--338, Pittsburgh, PA, June 1993.

[8] Jing, Y., Croft, W.B., 1994, An association thesaurus for information retrieval, in Proceedings of RIAO 94, 1994, pp. 146-160.

[9] Lu, A., Ayoub, M. and Dong, J. 1997. Ad hoc experiments using EUREKA. TREC-5, Pages 229-240.

[10] Mitra, M., Singhal, A. and Buckley, C., 1998, Improving Automatic Query Expansion. In Proc. of the 21st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp 206--214, Melbourne, August 24 - 28 1998.

[11] Qiu, Y. and Frei, H., 1993, Concept based query expansion. In Proc. of the 16th International ACM SIGIR Conference on R & D in Information Retrieval, pages 160--169. ACM Press, New York.

[12] Rocchio. J. 1971. Relevance feedback in information retrieval. The Smart Retrieval system---Experiments in Automatic Document Processing. G. Salton. Ed. Prentice-Hall Englewood Cliffs. NJ. pp.313-323.

[13] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. Modern Information Retrieval. Pearson Education Limited, England, 1999.

[14] Salton, G. and Buckley, C. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science. 41(4): pp. 288-297, 1990.

[15] Sparck Jones, K. 1971. Automatic keyword classification for information retrieval. Butterworths, London, UK.

[16] Wen, J.-R., Nie, J.-Y. and Zhang, H.-J. 2000. Clustering User Queries of a Search Engine. WWW10, May 1-5, 2001, Hong Kong.

[17] Xu, J. and Croft, W.B. 1996. Query expansion using local and global document analysis. In Proceedings of the 19th International Conference on Research and Development in Information Retrieval, pages 4--11, 1996.

[18] Xu, J. and Croft, W.B. 2000. Improving the effectiveness of information retrieval with local context analysis. ACM Transactions on Information Systems Vol.18, No.1, January 2000, Pages 79-11.