Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs

Chien-Kang Huang
Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, Republic of China. E-mail: ckhuang@mars.csie.ntu.edu.tw

Lee-Feng Chien
Institute of Information Science, Academia Sinica, Taipei, Taiwan, Republic of China. E-mail: lfchien@iis.sinica.edu.tw

Yen-Jen Oyang
Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, Republic of China. E-mail: yjoyang@csie.ntu.edu.tw

This paper proposes an effective term suggestion approach to interactive Web search. Conventional approaches to making term suggestions involve extracting co-occurring keyterms from highly ranked retrieved documents. Such approaches must deal with term extraction difficulties and interference from irrelevant documents, and, more importantly, have difficulty extracting terms that are conceptually related but do not frequently co-occur in documents. In this paper, we present a new, effective log-based approach to relevant term extraction and term suggestion. Using this approach, the relevant terms suggested for a user query are those that co-occur in similar query sessions from search engine logs, rather than in the retrieved documents. In addition, the suggested terms in each interactive search step can be organized according to its relevance to the entire query session, rather than to the most recent single query as in conventional approaches. The proposed approach was tested using a proxy server log containing about two million query transactions submitted to search engines in Taiwan. The obtained experimental results show that the proposed approach can provide organized and highly relevant terms, and can exploit the contextual information in a user's query session to make more effective suggestions.

1. Introduction

Identifying user information needs is always one of the most fundamental and challenging issues in the development of Web search engines. What makes this issue challenging is that most Web users give only short queries. Recent analyses of search engine logs revealed that the average length of Web queries is about 2.3 words (Jansen, 1998; Silverstein, 1998). Aimed at tackling the short query phenomenon in the Web environment, term suggestion mechanisms (Belkin, 2000) are commonly employed in search engine design (Altavista; Hotbot; Lycos).

Term suggestion is a kind of information retrieval technique that attempts to suggest relevant terms for user queries to help users formulate more effective queries and reduce unnecessary search steps. Different from query expansion techniques, which modify queries automatically, term suggestion techniques provide less active but more comprehensive aids for users if the suggested terms are highly related and well organized. Conventional approaches to term suggestion extract co-occurring key terms from retrieved documents that are ranked high (Anick, 1999; Sparck Jones & Stavely, 1999; Xu & Croft, 1996). These approaches are referred to as *document-based approaches* in this paper. Such approaches must deal with term extraction difficulties to ensure that the extracted keyterms are representative and have correct word boundaries in terms of the semantics. In addition, another problem of the document-based approaches is that the high-ranked documents might not all be relevant to the queries. Furthermore, as will be shown later in this paper, the document-based approaches cannot identify key terms that are conceptually related, but do not frequently co-occur in documents.

New alternative approaches of term suggestion identify relevant query terms in collected logs of user queries (Beeferman & Berger, 2000; Jansen, 1998; Nordlie, 1999, Ross & Wolfram, 2000). These approaches are referred to as *log-based approaches* in this paper. Beeferman and Berger

(2000) proposed an innovative query clustering method based on "click-through data." Each record of click-through data consists of a user's query to the search engine and the URLs that the user actually visits among the list provided by the search engine. By treating a click-through data set as a bipartite graph and identifying the mapping between queries and clicked URLs, queries with similar clicked URLs can be clustered. Based on a similar idea, Wen et al. (2001) conducted an experiment on encyclopedia search. The main problem with the approach based on click-through data sets is that a user normally only browses the high-ranked search results regardless of how long the list provided by the search engine is (Silverstein, 1998). As a result, most queries are associated with only a few URLs, and many URLs are not associated with any queries. This implies that the clustering results may be biased. Nevertheless, we agree with Beeferman and Berger (2000) who noted that "clustering queries submitted to search engines appears to be a rather less explored problem." Clearly, log-based approaches deserve further investigation.

In this paper, we propose a new, effective log-based approach to relevant term extraction and term suggestion. The first important feature of this approach is the development of a *query-session-based relevant term extraction method*. Using this method, the relevant terms suggested for original user queries are those that co-occur in similar query sessions from search engine logs, rather than in retrieved documents. A query session is defined as a sequence of a certain user's search requests submitted for a specific search subject. The second important feature of the proposed approach is the development of a *context-based term suggestion method.* When responding to a new request, existing term suggestion mechanisms usually fail to exploit the contextual information embedded in the query session that the current query request belongs to. In other words, when making term suggestions, all of the existing mechanisms look at only the terms submitted in the current query request and do not take into account the query terms submitted by the same user in prior requests. Using the proposed method, the suggested terms in each interactive search step can be obtained and organized according to their relevance to the whole query session, rather than using the most recent single query as in conventional methods. The following examples illustrate the basic ideas behind the proposed approach.

In the first example, there are users who want to find materials regarding "search engine technology." The three users may submit query sessions as follows:

first user: "search engine," "Web search," "Google"
second user: "multimedia search," "search engine," "Altavisita"
third user: "search engine," "Google," "Altavisita"

It is easy to see that some related terms, such as synonyms and alternative terms, may appear in a set of query sessions with similar requests. If the relevance between two query terms can be determined according to their co-occurrences and their associations with other query terms in query sessions, then query terms that are conceptually related can be extracted from collected query logs.

Session	Query Terms
Session 1	Obstetrics and Gynecology Department Children Hospital Taiwan University Hospital
Session 2	Taiwan University Hospital Medical College of Taiwan University Taiwan University Medical Library Journal Medial Journal
Session 3	Cathay General Hospital WanFang Hospital Taiwan University Hospital Tri-Service General Hospital

FIG. 1. Examples of contextual information in query sessions.

The second example demonstrates how the contextual information in query sessions can facilitate identifying users' demands. Figure 1 shows the query terms in the three query sessions submitted by three different users with different information demands. However, all three sessions contain the term "Taiwan University Hospital" ("台大醫院"). Through examining the entire sessions, one can easily figure out that the first user is looking for a hospital with a high-quality department of Obstetrics and Gynecology, while the second user wants to find some medical journals, and the third user wants to find a regional hospital in Taipei. On the other hand, it would be impossible to figure out the real demands of the users, if only the term "Taiwan University Hospital" was given.

With the two techniques discussed above, the proposed term suggestion approach can effectively exploit the contextual information in query sessions. The effectiveness of the proposed approach has been evaluated using a proxy server log with about two million query transactions submitted for Web searches. We have obtained promising experiment results so far. Compared with the document-based approaches, the relevant terms extracted using the proposed log-based approach are more relevant. More importantly, some conceptually related terms that do not often co-occur in documents could be extracted. In comparison with the conventional term suggestion mechanisms, the context-based term suggestion mechanism in the proposed approach can make more effective suggestions for an interactive Web search. However, the proposed approach has some weaknesses. It achieves lower recall in relevant term extraction and its term suggestion performance still depends on whether or not the collected log is adequate. Combining the proposed log-based approach with document-based approaches is therefore a subject for further research.

In the remainder of this paper, Section 2 reviews the related research and Section 3 presents an overview of the proposed term suggestion mechanism. Sections 4, 5 and 6

discuss three main issues concerning implementation of the proposed approach. Section 7 reports results obtained from experiments conducted to analyze the effectiveness of the proposed approach. Concluding remarks are given in Section 8.

2. Related Research

A recent survey of 40,000 Web users showed that, after a failed search, 76% of users try rephrasing their queries on the same search engine (NPD, 2000). This suggests that many users rely on search engines to help them formulate an optimal representation of their information needs. Information retrieval systems provide the desired function by invoking an interactive process of query reformulation and relevance feedback (Efthimiadis, 1996; Spink & Losee, 1996; Xu & Croft, 1996). Belkin (2000) characterized this type of interaction as *system-controlled* with respect to "term suggestion."

Term suggestion is an alternative way for the system to interact with the users. Given the terms used in the original query and/or the documents retrieved based on the original query, relevant terms that might be useful for query reformulation are suggested. It is the user's task in such a system to examine the suggested terms and to manually reformulate the query given the information provided by the system. Such techniques can be regarded as *user-controlled*, at least to the extent that the user controls how the query is reformulated. At Rutgers, researchers have been investigating support for query reformulation with respect to both relevance feedback versus term suggestion and to user knowledge and control of such support. Koenemann's results (1996) led researchers to draw the conclusion that explicit term suggestion is a more effective way with respect to query reformulation than automatic, behind-the-scenes query reformulation approaches. Especially in the Web domain, the outliers in the search results make the system-controlled approach even worse.

Thereafter, the short Web query and noisy search results characteristics have led researchers to investigate another alternative—clustering of queries. Clustering queries submitted to search engines appears to be a rather less explored alternative than clustering Web pages, though there are practical, commercial applications that exploit query clusters. For instance, if a user submits query q to a search engine, which is a member in cluster C, the search engine will suggest other members in C. Such "related query" tools are deployed in several search engines, including Lycos, AltaVista and HotBot. Nevertheless, earlier studies on query clustering were not based on the Web search environment (Brauen, 1971; Fitzpatrick & Dent, 1997; Raghavan & Sever, 1995; Wen et al., 2001). These query clustering methods were based on a small number of query and document collections. Query clustering employed in these investigations was to improve retrieval results. Term suggestion was not their main subject.

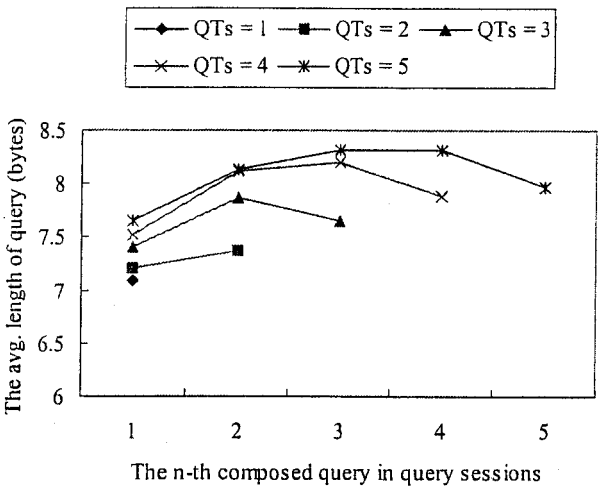

FIG. 2. The change of query length with respect to the positions of the queries in query sessions containing different numbers of composed queries.

Not only is the Internet much larger than the test documents used in previous research, the queries in the Web environment are much shorter than the test queries used in previous research. The average length of a TREC topic description for conventional text retrieval was 15 tokens (Voorhees & Harman, 1998), while a Web search engine log analysis revealed that the average query length for a Web search was about 2.3 tokens (Silverstein et al., 1998). Therefore, Web query clustering is very different from conventional query clustering, and as noted in the previous section, researchers have proposed relevant query clustering methods based on "click-through data" to discover correlations between queries and clicked URLs (Beeferman & Berger, 2000; Wen et al., 2001).

3. Overview of the Proposed Approach

Before describing the proposed approach in detail, we will discuss an observation on a query session log containing 615,634 query sessions for Web search. The details of the log will be introduced in the next section.

The observation is presented in Figure 2, which depicts how the average length of a query in numbers of bytes varies with the position of the query in query sessions. The folder lines marked by QTs = n correspond to query sessions containing exactly n queries. The x-axis corresponds to the position of a query in the query session, and the y-axis corresponds to the average length of the query in bytes. As most of the queries collected in the log are in Chinese, the length roughly indicates the number of Chinese characters, which was found to be around three to four (approximately equivalent to two words in English). Note that each Chinese character occupies two bytes in the computer, and that most Chinese words contain one or two characters.

Figure 2 reveals several important characteristics of query sessions. First, the average length of a query term

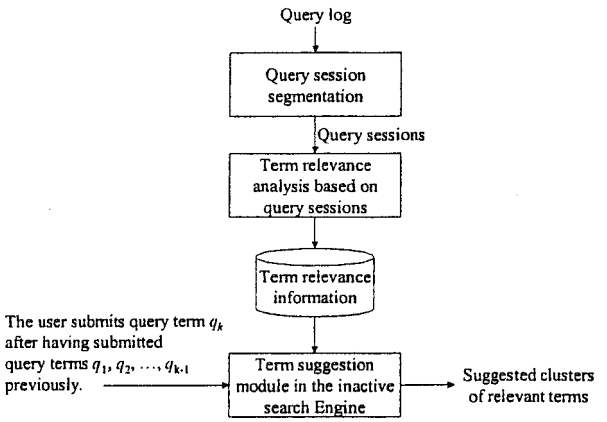

FIG. 3. The basic operations of the proposed term suggestion mechanism.

appearing in the later part of a query session generally is greater than that of a query term appearing at the beginning. Furthermore, close examination of the query terms appearing in the same query session reveals that the first query terms are often too general to satisfy search requests. The query terms appearing in the latter part of a query session generally are more effective than the query terms appearing the earlier part. This observation strongly suggests that effective term suggestions are required especially for the first query terms, and that contextual information embedded in query sessions is worth exploiting to facilitate users' search efforts.

To facilitate the use of contextual information in query sessions to make effective term suggestions, this paper proposes an effective log-based approach. Figure 3 depicts the basic operations of the proposed term suggestion mechanism. The kernel term suggestion module operates based on a term relevance analysis conducted in advance on a collected log of user queries. The query log, containing query transactions submitted to search engines, is first partitioned into a number of query sessions. The relevance among the query terms in the log is computed based on how these query terms are clustered in query sessions. The result from the term relevance analysis is then exploited by the term suggestion module on-the-fly. When a user submits a new query term q_k after having submitted a series of query terms $q_1, q_2, \ldots, q_{k-1}$ previously, the term suggestion module will suggest and organize terms that are relevant not only to the currently submitted term q_k but also to the previously submitted terms $q_1, q_2, \ldots, q_{k-1}$.

To develop an interactive search engine that incorporates the proposed term suggestion mechanism, the following three issues must be addressed.

1. Partitioning the query log into query sessions accurately. In other words, the boundary of each query session must be determined.
2. Measuring the relevance between each pair of query terms in the query session log and extracting relevant query terms for each input query.
3. Exploiting term relevance and contextual information from a query session to make effective and organized term suggestions on the fly.

These three issues will be elaborated in the following sections.

4. Query Session Segmentation

The first task in this type of research is to collect an appropriate query session log for analysis. A proxy server of a regional network center easily contains thousands of clients that use it to access the Web. Not only do the search requests from the clients pass through the proxy server, but also all of the HTTP requests are the same. Compared with common search engine logs, a proxy server's log can record richer information regarding users' information access and, more importantly, the recorded search requests are not limited to certain search engines. A method for query session segmentation from a proxy server log has, therefore, been developed.

The major challenge for an effective query session segmentation method is instantly determining the starting, subsequent and ending requests of a query session from a huge number of recorded search requests received from users. It is fortunate that, as we found in our experiments, a few segmentation errors did not affect very much the performance of the proposed approach for relevant term extraction. Also, it has been reported that most search requests possess the time locality property, and that most users submit few requests per day (Silverstein et al., 1998). Silverstein et al. (1998) claimed that queries for single piece of information come clustered in terms of time, and that then there is a gap before the user returns to the search engine. Based on this observation, we use a time threshold as a delimiter to segment query sessions.

For each query transaction T_i in a proxy server log we examine the following three pieces of information:

1. t_i: timestamp at which the query transaction was submitted;
2. id_i: the IP address of the machine at which the query transaction was submitted;
3. q_i: the URL string in the query transaction.

We then define a query session as follows:

Definition: *A query session S* is a sequence of query transactions (T_1, T_2,. . .,T_n), which satisfies the following conditions:

1. $id_1 = id_2 = \ldots = id_n$;
2. $t_{i+1} - t_i <$ time threshold, for all 1 < i &lteq; n − 1, where the time threshold is a parameter to be set;
3. no other query transactions can join *S* to form a larger query session.

The time threshold is, however, the major parameter needs to be well tuned.

TABLE 1. The statistics of the query log used for analysis.

Time span of the log	126 days (2000/4/26 ~ 2000/9/5)
Number of clients	21,421
Number of query transactions	2,369,282
Number of distinct query terms	218,362

To test the performance of the above segmentation method, several experiments have been performed with a query log obtained from a local proxy server at National Taiwan University, which served 52 organizations, including 20 universities and colleges in northern Taiwan. The query log contained all of the search requests from 21,421 clients submitted to general-purpose Web search engines, including Yahoo-Taiwan, Sina-Taiwan, Pchome and Yam, which are major players in Taiwan. During a logging session of 126 days, we collected 2,369,282 query transactions for analysis. Table 1 shows the statistics of the query log collected.

Figure 4 shows how the number of segmented query sessions that contain more than one query transaction varied with different time threshold values. We are interested in query sessions containing more than one query transaction because the following term relevance analysis is based on these long query sessions. Since the curve shown in Figure 4 saturates around 300 seconds, we therefore selected 5 minutes as the time threshold. It is interesting that the same 5-minute time threshold was also used by Silverstein et al. (1998). Table 2 shows the percentage of query sessions containing more than one query, if the time threshold is set to 5 minutes. It is also interesting to find that the percentage is very close to that reported by Silverstein et al. (1998).

One can imagine that segmentation cannot be done flawlessly. To determine the segmentation performance, we randomly selected a set of 1,000 segmented query sessions

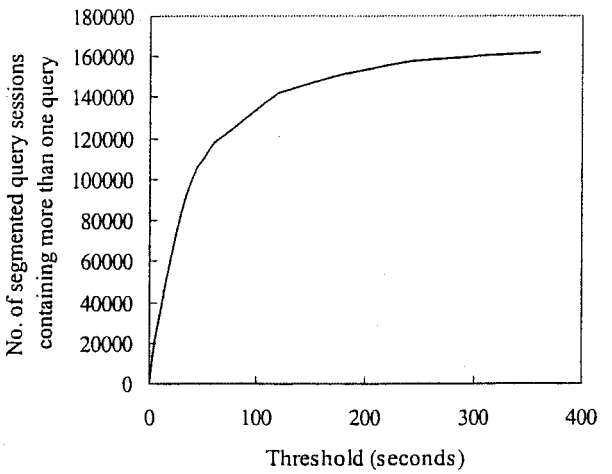

FIG. 4. Curve showing how the number of segmented query sessions varies with different time threshold values, where only the query sessions containing more than one query are counted.

TABLE 2. The percentage of query sessions containing more than one query, if the time threshold is set to 5 minutes.

	Number of obtained query sessions	Ratios
Sessions containing single query	455,454	74%
Sessions containing more than one query	160,180	26%
Total	615,634	100%

and manually examined whether the composed queries in a session contained non-relevant terms or whether relevant terms were mis-segmented in consecutive sessions. Examination revealed that more than 97% of the segmentations agreed with human intuition. This high quality of the segmentation results was due to the fact that most of the users submitted only one search request in a day, and that for requests with more than one query, the time that elapsed between consecutive queries was often short. In other words, only a small portion of the users' query transactions had multiple search requests in a day and could cause segmentation errors. To illustrate, Figure 5 shows the average numbers of composed queries of the segmented query sessions, which were obtained with different thresholds. We found that most of the users submitted only one search request in a day, and that on average a request contained less than two queries. High segmentation accuracy could be achieved simply by using the IP address and time threshold. In fact, a mis-segmented query term did not affect very much the performance of the proposed approach to extracting relevant terms unless it co-occurred with the same but non-relevant query terms in a mis-segmented session many times.

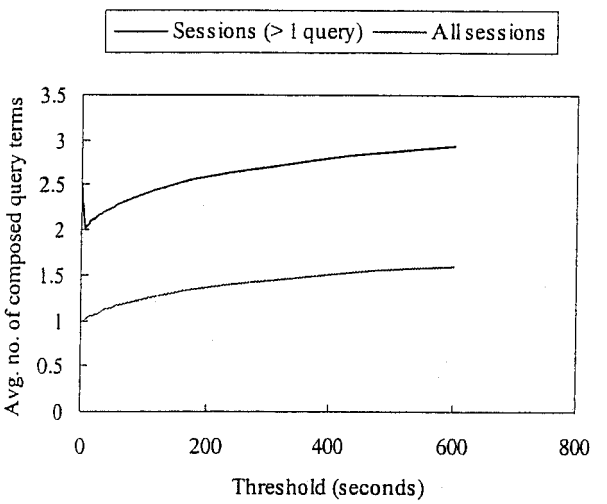

FIG. 5. Curves show the average numbers of composed queries in the segmented query sessions, which varied with different time threshold values. The upper curve depicts the results for sessions with more than one query, and the lower curve for the results for all the sessions.

```
function compute_relevant_term_set(q_u, Q, C, R)
{
  Input:
    q_u: the query term of concern
    Q: the set of all query terms in the log
    C: the co-occurrence matrix
  Output:
    R: relevant term set
  R = ∅
  For every q_v in Q {
    if ( C_{u,v} ≥ √f_u )
      R = R ∪ {q_v}
    else if ( ∜f_u ≤ C_{u,v} ≤ √f_u )
      if (f_u >> f_v or f_u << f_v)
        if (Dependence(q_u, q_v) > threshold_1)
        then R = R ∪ {q_v}
      else
        if (Jaccard(q_u, q_v)> threshold_2)
        then R = R ∪ {q_v}
    else if ( C_{u,v} ≤ ∜f_u ) {
      If (cos(q_u, q_v) > threshold_3)
      then R = R ∪ {q_v}
  }
  return R;
}
```

FIG. 6. The algorithm for computing the relevant term set.

5. Measure of Term Relevance

With query sessions segmented, the next step is to perform relevance analysis of query terms. The relevance analysis proposed in this paper is based on a term co-occurrence matrix defined as follows.

Definition: *The co-occurrence matrix* **C** *of the distinct query terms in a query log, denoted by* $\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_n}$, *is an* **n** *by* **n** *matrix with* $\mathbf{C_{i,j}}$ = *the number of query sessions containing both query terms* $\mathbf{q_i}$ *and* $\mathbf{q_j}$. *(let* $\mathbf{fi} = C_{i,I}$, *which denotes the number of query sessions containing* q_i*).*

Three notable similarity estimation functions are applied in our relevance analysis—the Jaccard measure, dependence measure and Cosine measure, which are defined below (Manning, 1999):

$$\text{Jaccard}(q_u,q_v) = \frac{C_{u,v}}{f_u + f_v - C_{u,v}}, \tag{1}$$

$$\text{Dependence}(q_u,q_v) = \frac{C_{u,v}}{\min(f_u,f_v)}, \tag{2}$$

$$\cos(q_u,q_v) = \frac{\sum_{\forall q_j}(C_{u,j} \cdot C_{v,j})}{\sqrt{\sum_{\forall q_j} C_{u,j}^2} \cdot \sqrt{\sum_{\forall q_j} C_{v,j}^2}}. \tag{3}$$

The Jaccard measure represents the relative frequency of co-occurrence. The experimental results described in the next section will show the Jaccard measure is useful when q_u and q_v contain high co-occurrence frequencies. The dependence measure is a degenerated form of the Jaccard measure. When the difference between the frequencies of q_u and q_v is large, the larger item from the original formula will be eliminated in order to deal with low-frequency query terms. However, co-occurrence analysis is applicable to higher frequency queries because higher frequency queries are more likely to appear with other query terms. On the other hand, lower frequency terms have little chance of appearing with other query terms in sessions. The Cosine measure is thus adopted to deal with this problem. As similar query sessions share similar terms, for each query, we take the candidate relevant terms as its feature vector. The similarity between queries can be computed from their feature vectors. Thus, lower frequency terms still have a chance to extract relevant terms.

Figure 6 shows the relevant term extraction (RTE) algorithm that we implemented. One may note that the algorithm does not apply a global formula to all cases. This practice is due to the empirical experiences to be further discussed in the next section. The applied measure functions in the algorithm depend on three different degrees of co-occurrence conditions between q_u and q_v: high co-occurrence, medium co-occurrence and low co-occurrence. When the co-occurrence value of q_u and q_v is between $\sqrt[4]{f_u}$ and $\sqrt{f_u}$, either the dependence measure or the Jaccard measure will be applied. If the difference between the frequencies of qu and qv is large, we apply the dependence measure; otherwise, we apply the Jaccard measure. Queries that frequently co-occur with query q_u, i.e., $C_{u,v} \geq \sqrt{f_u}$, are taken as being relevant to query q_u. For queries which seldom co-occur with query q_u, i.e., $C_{u,v} \leq \sqrt[4]{f_u}$, we apply the cosine measure, to obtain more less-relevant query terms. As will be shown in the next section, the required parameter values, including, $\sqrt[4]{f_u}$, $\sqrt{f_i}$ and *threshold1*$\sim$*3*, were obtained through extensive experiments.

6. Relevant Term Extraction

To achieve better performance with the proposed RTE method, three sets of experiments were designed. The first set of the experiments was performed to obtain appropriate parameters for the proposed method, the second set to extract relevant terms, and the third set to compare the achieved performance with that of conventional document-based methods. The segmented query session log described in Table 2 was used in the experiments. Some further statistics of the log are listed in Table 3.

In the experiments, there are 160,180 sessions that contain more than one query. In the query sessions, there are 5,366 distinct query terms occurring over 10 times. We randomly selected 95 query terms as the test query set among them. The number of occurrence of the test query terms ranges from

TABLE 3. Some statistics for the log used for relevant term extraction.

	All query sessions	Sessions with more than one query
Number of obtained query sessions	615,634	160,180
Total number of query terms in the sessions	2,369,282	1,213,226
Number of distinct query terms	218,362	177,324
Average number of distinct query terms per session	1.45	2.75

1,054 times for "MP3" to 10 times for a company's name. About half of the test query terms are proper nouns, such as Web site names, company names and personal names, and the remaining are mostly subject terms such as price of flight ticket, titles of music and names of theaters. We adopted the F_b-measure defined below as our evaluation metric.

$$F_b = \frac{(b^2 + 1)pr}{b^2p + r}, \quad (4)$$

where p is precision, r is recall and b is a specified parameter that reflects the relative importance of recall and precision. Both F_1 (b = 1) and F_2 (b = 2) were applied in our evaluation because we were more interested in precision than recall in the relevant term extraction process for Web search. The metrics were used to observe the performance of the extracted relevant terms under different parameter and threshold value settings. The recall set for each test query was obtained through manual analysis from all of its co-occurring terms and the terms that co-occurred with their neighbors. Five volunteers participated in the analysis. A relevant term had to be judged as relevant to the query by at least three of the volunteers. Since the log size was not large enough, the obtained recall sets were only used as a reference for parameter settings.

Experiments on Parameter Settings

The diagram in Figure 7 shows the training process for parameter settings.

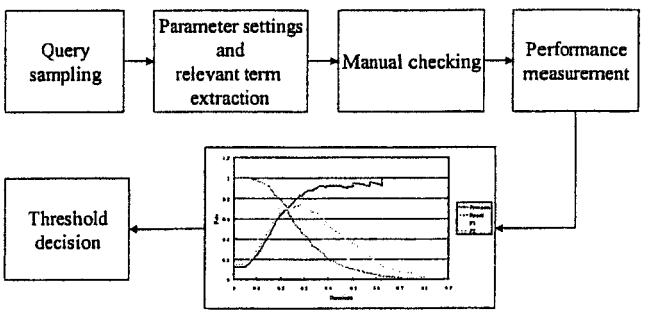

FIG. 7. The training process for the threshold and parameter settings adopted in the relevant term extraction method.

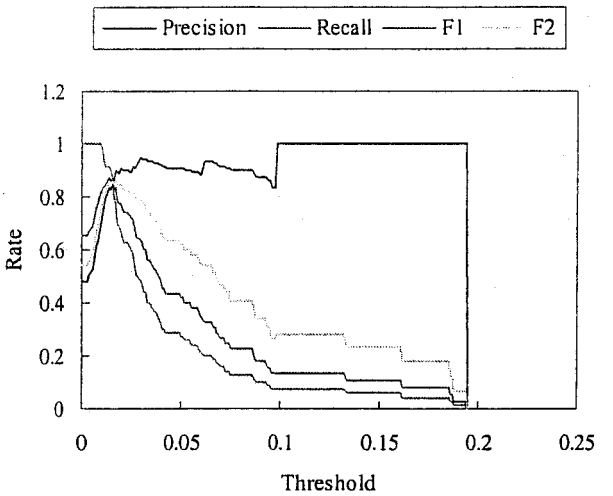

FIG. 8. The performance of the Jaccard measure function in dealing with the high co-occurrence case.

The purpose of the first experiment was to evaluate the performance of the Jaccard measure function in dealing with the high co-occurrence case. Figure 8 shows the performance obtained. It is noted that the precision obtained could be very high even when the $threshold_2$ was lowered to 0.015. According to the experimental results, the highest value of F_1-measure appeared when the threshold was 0.013, and the highest value of F_2-measure appeared when the threshold was 0.017. The Jaccard measure function was especially useful in dealing with the high co-occurrence case.

The purpose of the second experiment was to evaluate the performance of the dependence measure in dealing with the medium co-occurrence case, but, in this case, the difference between the frequencies of q_u and q_v was large. Figure 9 shows the performance obtained. It is noted that the

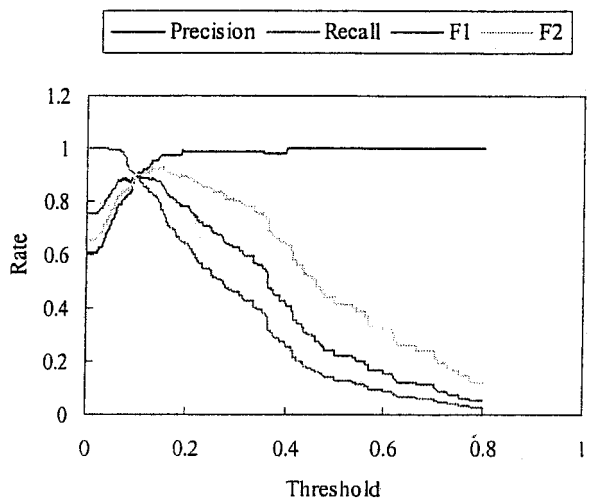

FIG. 9. The performance of the dependence measure function in dealing with the medium co-occurrence case when the difference between the frequencies of q_u and q_v was large.

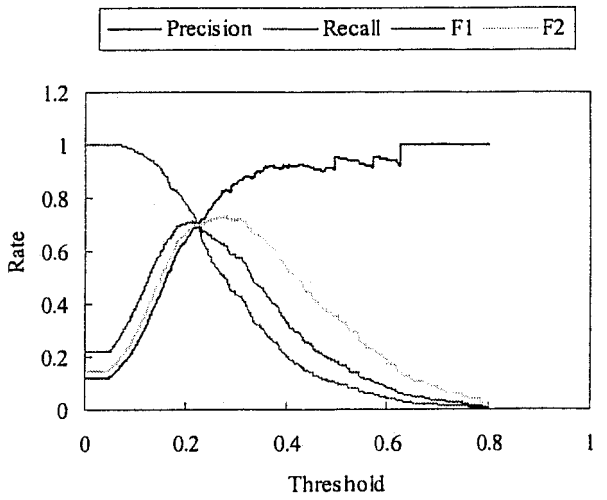

FIG. 10. The performance of the of the Cosine measure with vector space analysis in dealing with the low co-occurrence case.

precision obtained was also high in this case even when the threshold was lowered to 0.15. According to the experimental results, the highest value of F_1-measure appeared when the *threshold1* was 0.103, and the highest value of F_2-measure appeared when the *threshold1* was 0.147. The dependence measure function was effective in dealing with the special medium co-occurrence case.

The purpose of the third experiment was to evaluate the performance of the Cosine measure with vector space analysis in dealing with the low co-occurrence case. Figure 10 shows the performance obtained. It is interesting to note that the obtained curves are much smoother than those in the previous two experiments. According to the experimental results, the highest value of F_1-measure appeared when the *threshold3* was 0.216, and the highest value of F_2-measure appeared when the *threshold3* was 0.276. With the Cosine measure, some relevant terms that could not be extracted with the previous functions could now be derived for those low-frequency search terms, and many less-relevant search terms could also be extracted to increase the recall rate.

Based on the above experiments, we can determine a set of appropriate thresholds. Table 4 shows the recall and precision rates for the cases with the highest F_1-measure and F_2-measure values, respectively. It should be noted again that the obtained recall rates are a reference for parameter settings only.

Experiments on Relevant Term Extraction

Based on the thresholds that achieved the highest F2-measure values, the proposed method was further performed to extract relevant query terms for the set of 5,366 distinct query terms that occur over 10 times in the log. It successfully extracted relevant terms for 3,330 of them. On average there were 9.28 relevant terms extracted for high-frequency terms, 4.82 for medium-frequency terms, and 2.77 for low-frequency terms. Though the test log size was small, the extracted terms, especially the high-frequency query terms, were highly relevant and would have been hard to obtain through manual analysis. The achieved relevance between the query terms and extracted relevant terms exceeded our expectations.

Comparison with the Document-based Method

To compare the performance of the proposed log-based RTE method with that of the conventional document-based methods, especially with regard to the precision rate for the extracted terms, further experiments were performed. Conventional IR systems often rely on key terms extracted from retrieved documents. In our experiments, a document-based method was implemented in combination with the Google Chinese search engine. Our basic idea in using the document-based method was to extract co-occurring key terms for each test query from a set of Web documents D, that were the high-ranked documents retrieved from Google Chinese. To reduce the term extraction difficulties, it was assumed that a set of basic key terms W existed and had been collected. A query log from Dreamer, a representative search engine in Taiwan, was collected as the basic key term set for analysis. The Dreamer log contained 228,566 distinct query terms and the total number of query transactions was 2,184,256. We took the top 20,000 query terms as the basic key term set, which represented 81% of the queries submitted. However, use of the predefined term set still limited the recall rate of the document-based method, so it would be better to combine it with keyword extraction methods if a real application is to be developed based on document-based methods.

Measurement of term relevance was based on a mutual-information-based association measure between each query term t and its candidate term w. The w was assumed to appear in D and belong to W. This satisfied the condition that $N(w,t)/N(w)+N(t) >$ a threshold value, in which $N(w)$, $N(t)$, and $N(w,t)$ were the numbers of Web documents in D containing term w, containing term t, or containing both term w and t, respectively. In our experiments, we collected up to 100 search result entries for each test query term and extracted each entry's title and description as the representation of the corresponding document containing the query term. Since the title and description information can be obtained directly from the search results returned by search engines, it is possible to avoid retrieving a number of additional pages that are required if we include the contents of the retrieved documents in the processing stage. This would significantly reduce the computing time and would

TABLE 4. The obtained recall and precisions for the cases with the highest F_1-measure and F_2-measure values.

	Recall	Precision
The obtained highest F2-measure	0.648	0.890
The obtained highest F1-measure	0.822	0.772

TABLE 5. Performance comparison between the document-based method and the query-session-log-based method on the top-10 extracted terms and the overall term sets, respectively.

	The document-based Method	The log-based method
Top-10 precision	0.46	0.81
Top-10 recall	0.15	0.29
Top-10 average relevance	1.85	2.42
Overall precision	0.26	0.90
Overall recall	0.66	0.45
Overall average relevance	1.73	2.40
Average Number of extracted terms and truly relevant terms	89.5/22.8	11.8/10.6
Comments	Relatively low precision	High precision
	High recall	Low recall
	Useful for low frequency queries	More comprehensive
		Needs sufficient log

be efficient for on-line term suggestion, but the recall rate for the extracted relevant terms might not be good enough in this case.

The test query set contained 100 queries randomly selected from 3,330 queries whose relevant query terms had been extracted using the proposed log-based method. For each of the test queries, the document-based relevant terms were also extracted using the implemented document-based method for comparison. The relevant terms extracted using the two different methods were merged and inspected manually to judge the relevance between the test query terms. For each test query term, its relevant term set was defined as consisting of all of the relevant terms extracted using the two different methods. To obtain a higher recall rate for the extracted relevant terms, the threshold used in the document-based method was relaxed. The average numbers of extracted terms and of truly relevant terms obtained were 89.5 and 22.8 using the document-based method, and 11.8 and 10.6 using the log-based method, respectively.

In addition to the precision and recall rates, we also applied a new measure called the average relevancy to those extracted relevant terms. When the volunteers judged the relevance between the test query terms and the extracted relevant terms, they were asked to assign numeric values ranging from 0 to 3 to indicate the degree of relevance. A value of 0 meant non-relevant, and a value of 3 meant highly relevant. The average relevancy was, then, the average relevancy value of the extracted relevant terms. Table 5 shows the obtained performance on the top-10 extracted terms and the overall term sets, respectively.

It is clear that the proposed log-based method performed better than the document-based method in terms of the precision rate. The unsatisfactory precision rate of the document-based method might have resulted from the use of a predefined vocabulary set, and from the low relevancy of the retrieved documents. In fact, the above experiment was mainly performed to find out whether the relevant terms extracted using the log-based method would be highly relevant, and whether the log-based method could extract relevant terms that would be conceptually related but would not frequently co-occur in the same documents. The experimental results have shown that the proposed log-based method can achieve very high average relevancy. Although relevancy could not be evaluated without some subjectivity, we still have strong confidence in the results obtained. This is because the proposed method achieved better average precision rates for almost every test query, and because among the on average 11.8 relevant terms extracted by the proposed method, 7.2 terms of them could not be found using the document-based method. Significantly, close examination performed manually showed that only a few of them occurred in the retrieved documents.

However, the log-based method was not superior in every respect. The number of extracted relevant terms was obviously less than that for the document-based method. The Web contains a huge number of pages. For most test queries, the document-based method can extract more relevant terms than the log-based method. The proposed method also required a sufficiently large log. Considering that the two different methods are complementary in many aspects, how to combine them to deal with the relevant term extraction problem is worthy of further investigation.

Figure 11 illustrates the test query 新浪 (Sina in Chinese; Sina is a Chinese Web portal) as an example to show the characteristics of the two different methods. The relevant terms extracted using the document-based method were the most popular services provided by Sina. No names from Sina's portal competitors in Taiwan such as Yahoo, Pchome, Yam, etc. were extracted. This result was probably due to the fact that the top relevant pages retrieved from Google were limited to Sina's sites. In fact, in our observations many relevant terms are conceptually related, but do not frequently co-occur in the same documents. Yahoo is a

The proposed log-based method	The document-based method
新浪網 (Sina)	**新浪網 (Sina)**
pchome	著作權 (copyright)
yahoo	智慧財產權 (IP right)
kimo	聊天 (chat)
sex	新聞 (news)
yam	Nasdaq
雅虎 (Yahoo)	聊天室 (chat room)
tomail	理財 (personal finance)
免費信箱 (free email)	金融 (banking)
中文雅虎 (Yahoo Chinese)	華淵 (company name)
奇摩 (Kimo)	信箱 (mail box)
搜尋引擎 (search engine)	翻譯 (translation)
	算命 (fortune telling)

FIG. 11. An example showing the different characteristics of the relevant terms extracted using the log-based and document-based methods, respectively, in which the test query was 新浪 (Sina in Chinese).

typical example; it might occur frequently with Sina in news stories but seldom co-occur in the same Web pages. To extract conceptually related terms, such as the names of competitors is an important merit of the proposed log-based method.

7. Context-based Term Suggestion

Once the relevance between each pair of query terms in a query log has been computed, the interactive search engine has all the background information it needs to start operating. In this section we will propose a term suggestion algorithm based on the relevant term extraction method described in Section 5. As mentioned earlier, the major contribution of the proposed term suggestion mechanism is that it exploits the contextual information embedded in the query sessions that is currently being handled.

First, we organize the suggested terms through the following *term organization process.*

1. *Collect all relevant terms*. Apply the algorithm shown in Figure 6 to collect all relevant terms from query sessions. Assume that q_u is the query of interest, R is the set of all obtained relevant terms of q_u, and that q_v is one of the relevant terms. Continue with steps 2 and 3 below for each q_v in R.
2. *Extract highly relevant terms.* If Jaccard(q_u, q_v) $>$ threshold$_4$, or $C_{u,v}/f_v > threshold_5$ then label q_v as a "highly relevant" term of q_u, and remove it from the term set R.
3. *Cluster the remaining relevant terms.* In this step, we organize the relevant terms remaining in R according to different search concepts. To divide the relevant terms into different clusters, a clustering approach is applied. For each q_v , q_u's relevant term, we take its candidate relevant terms obtained from the query session log as q_v's feature vector, and take the frequencies of co-occurrence as q_v's feature values. The similarity between each pair of q_u's relevant terms can be computed with a Cosine measure from the corresponding feature vectors. The relevant terms with similar features are grouped into clusters. We apply the single-linkage hierarchical clustering algorithm to estimate the similarity between clusters (van Riemsdijk, 1979). The clustering procedure estimates the similarity between all pairs of composed relevant terms and uses the highest term similarity value as the cluster similarity value. The two most similar clusters are merged to form a new cluster, and the original clusters are removed. The above procedure is repeated until the similarity between any two clusters is smaller than $threshold_6$. The remaining clusters are taken as the result.

The above term organization process will produce two types of relevant terms, i.e., the highly relevant terms and the relevant terms that are clustered. In our experience, the highly relevant terms are often synonyms of the query terms. As for the relevant terms grouped in the same clusters, they are mostly the terms expressing similar requests.

Then, *the term suggestion process* can be performed. Basically, this process re-ranks and filters the relevant terms of the current query term $\boldsymbol{q_k}$ by estimating the vector-space-based relevance between each pair of relevant terms and previous query terms $\boldsymbol{q_1, q_2, \ldots, q_{k-1}}$. The terms that the process suggests are not only relevant to $\boldsymbol{q_k}$ but also relevant to some of the previous query terms. The procedure performed in the term suggestion process is described below.

1. Assume that q_k is the current query term submitted by the user, $q_1, q_2, \ldots, q_{k-1}$ are the previous query terms. Assume also that S is the set of all clusters of relevant terms of q_k, which is generated by the above term organization process, and that q_r is a relevant term appearing in S.
2. For each q_r in S, calculate the contextual Cosine measure. The contextual cosine coefficient is defined as equation (5), in which the Cosine measure is defined similar to equation (3):

$$\begin{aligned}\text{contextual\_cos}(q_r, q_1, q_2, \ldots, q_{k+1}) &= \cos(q_r, q_{k+1}) + \alpha \cos(q_r, q_{k-2}) \\ &\quad + \cdots + \alpha^{k-2}\cos(q_r, q_1). \quad (5)\end{aligned}$$

where α is a value between 0 and 1. The α coefficients are weighting values used to adjust the significance of each item of Cosine value based on its distance from the query of concern. If the contextual_cos(q_r, $q1$, . . ., q_{k-1}) $<$ $threshold_7$, then q_r is removed from the corresponding cluster in S. If the corresponding cluster does not contain any relevant terms, then the cluster is removed from S.

3. Re-rank the relevant terms in each cluster according to their contextual Cosine measures.
4. Re-rank all the clusters in S according to the average contextual Cosine measure of each cluster. Assume that CL is a cluster in S. The average contextual Cosine measure is defined as equation (6):

$$\begin{aligned}\text{avg\_contextual\_cos}(CL, q_1, q_2, \ldots, q_k) &= \frac{1}{|CL|}\sum\nolimits_{q_r \in CL} \text{contextual\_} \\ &\quad \cos(q_r, q_1, q_2, \cdots, q_k). \quad (6)\end{aligned}$$

In the above procedure, steps 3 and 4 are performed to re-rank the suggested terms and the clusters of terms. All the highly relevant terms and the final S are sorted according to their relevance and suggested to the user that submitted the query. In the next section, we will discuss the effectiveness of this empirical algorithm.

Experimental Results

We conducted some experiments to determine the effectiveness of the proposed term suggestion mechanism. The

TABLE 6. An example showing the terms suggested by the proposed mechanism and how they were organized into clusters with similar search concepts and relevance to the entire query session.

Query sessions	Term suggestion
1. NTU	Suggested highly relevant terms for "NTU" NTU Hompage, Taiwan University Suggested clusters of relevant terms for "NTU" 1. Library, NTU Library, National Library, Journal 2. Language Training and Testing Center, NTNU Language Training Center 3. NTU Computer Center, NTU EE 4. NTU Hospital, NTU Medical College, Library of NTU Medical College, Medical Journal, NTU Medical Department 5. University, Campus 6. NCKU, YMU, NCCU, TKU, FJU, SCU 7. NTU BBS, Palm BBS 8. Entrance Examination, Admission Announcement
1. Library 2. NTU	Suggested highly relevant terms for "NTU" NTU Hompage, Taiwan University Suggested clusters of relevant terms for "NTU" 1. NTU Library, National Library, Journal 2. Medical Journal 3. Language Training and Testing Center
1. Hospital 2. NTU	Suggested highly relevant for terms "NTU" NTU Hompage, Taiwan University Suggested clusters of relevant terms for "NTU" 1. NTU Hospital, NTU Medical College, NTU Medical Department, Library of NTU Medical College, Medical Journal

log used for term suggestion was the same as the query log used in the term relevance experiment. We listed the obtained different term suggestion results for the same query term used in different sessions in Table 6. The first row presents an example that shows the clusters of relevant terms suggested by the proposed mechanism when it was given only one query term, "NTU" (an abbreviation for National Taiwan University). The following rows show the suggested terms produced in response to different previous queries, "library" and "hospital." It should be noted that all of the illustrated queries and suggested terms were translated into English from Chinese.

The illustrated example is not an extreme case. In our experiments, many high frequency queries could produce appropriate suggestions. A quantitative experiment was also conducted to evaluate the effectiveness of the proposed mechanism at reducing the number of query requests the user needed to input to get the desired information. In the experiment, each query session in our log was fed into the proposed mechanism. It should be noted that the occurrence of each test query session was not considered in the process of determining its relevant term set. That is to say, each of the test query sessions was actually unknown to the term suggestion mechanism at the step in which its relevant terms were determined. If one of the terms suggested by the proposed mechanism in one step of the query session appeared in a later step of the query session, it was considered that the proposed mechanism had made a successful term suggestion. Table 7 shows the experimental results obtained after analysis of 160,180 query sessions. The statistics reveal that the proposed mechanism made successful term suggestions in 20.4% of the query sessions and could reduce the average number of transactions that the user submitted in one session from 2.75 to 2.3.

8. Concluding Remarks and Future Research Issues

Identifying user information needs is one of the most fundamental and challenging issues in the development of Web search engines. What makes this issue challenging is that most Web users provide only short queries. This short query problem has led to the development of term suggestion mechanisms.

Conventional approaches for making term suggestions extract co-occurring key terms from retrieved relevant documents. This paper presented a novel, effective log-based approach for performing relevant term extraction and term suggestion. Using this approach, the relevant terms suggested in response to users' original queries are those that co-occur in similar query sessions of search engine logs, rather than from retrieved documents. Many relevant terms that are conceptually related but do not frequently co-occur in the same documents can, therefore, be identified.

This approach also exploits the contextual information embedded in a series of query terms submitted by a user in a search process. Exploiting the contextual information helps the search engine identify the user's exact needs. The suggested terms in each interactive search step can be identified and organized according to their relevance to the entire query session, rather than only to the most recent single query as in conventional approaches. The proposed approach was evaluated with a proxy server log with about two million query transactions submitted for Web search. The experimental results show that the proposed approach

TABLE 7. The reduction in the number of query requests achieved with the proposed term suggestion mechanism.

Total number of query sessionscontaining 2 or more query requests	Number of sessions with successful term suggestions	Average number of suggested terms foreach query requests	Average number of requests in a query session	Average reduction in the number of query requests
160,180	32,665 (20.4%)	15.87	2.75	0.45

can provide organized and comprehensive relevant terms, and can effectively exploit the contextual information in users' query sessions to make more effective suggestions.

However, the proposed log-based approach is not superior in every respect. The number of extracted relevant terms is obviously less than that for the document-based method. The log-based term suggestion mechanism is not as strong in dealing with low frequency query terms. More importantly, the proposed log-based approach requires a sufficiently large log. Though the experimental results are promising, there are issues that deserve further study:

1. How can the log-based and document-based methods be combined to deal with the relevant term extraction problem.
2. How can term suggestion be performed for low frequency query terms.
3. How can the suggested terms be organized so as to conform with the concept hierarchies of human beings.

Regarding the first two issues, we have been investigating how to integrate the log-based approaches and the conventional document-based approaches so that the merits of both types of approaches are exploited. As for the third issue, we will investigate the effects of alternative clustering algorithms and similarity measurements.

References

AltaVista Inc. http://www.altavista.com

Anick, P.G., & Tipirneni, S. (1999). The paraphrase search assistant: terminology feedback for iterative information seeking. In Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'99), 153–159.

Beeferman, D., & Berger, A. (2000). Agglomerative clustering of a search engine query log. In Proceeding of International ACM SIGKDD Conference on Knowledge (ACM SIGKDD'00).

Belkin, N.J. (2000). Helping people find what they don't know. Communication of ACM (CACM), 43(8), 58–61.

Brauen, T.L. (1971). Document vector modification. In G. Salton (ed.), The SMART retrieval system: experiments in automatic document processing. Englewood Cliffs, NJ: Prentice Hall, 456–484.

Chuang, S.-L., & Chien, L.-F. (2003). Automatic subject categorization of query terms for web information retrieval. Special Issue on Web Mining and Retrieval, Journal of Decision Support Systems, 2003.

Efthimiadis, E. (1996). Query expansion. Annual Review of Information Science Technology, 31, 121–187.

Fitzpatrick, L. & Dent, M. (1997). Automatic feedback using past queries: social searching? In Proceedings of 20th International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'97), 306–313.

Hotbot Inc. http://www.hotbot.com

Jansen, B.J., et al. (1998). Real life information retrieval: A study of user queries on the web. In SIGIR FORUM, 32(1), 5–17.

Koenemann, J. (1996). Relevance feedback: usage, usability,utility. Ph.D. Dissertation, Rutgers University, Dept. of Psychology.

Lycos Inc. http://www.lycos.com

Manning, C. & Schutze, H. (1999). Foundations of statistical natural language processing: 8. Lexical acquisition. Cambridge, MA: MIT Press.

Nordlie, R. (1999). User revealment—a comparison of initial queries and ensuing question development in online searching and in human reference interaction. In Proceedings of 22th International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'99), (pp. 11–18).

NPD Search and Portal Site Survey. Published by NPD New Media Services. (2000). Available: http://www.searchenginewatch.com.

Raghavan, V.V., & Sever, H. (1995). On the reuse of past optimal queries. In Proceedings of the 18th International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'95), (pp. 344–350).

Ross, N.C.M., & Wolfram, D. (2000). End user searching on the Internet: an analysis of term pair topics submitted to the excite search engine. Journal of the American Society of Information Science, 51, 949–958.

Silverstein, C., et al. (1998). Analysis of a very large AltaVista query log. Technical Report 1998–014, Digital Systems Research Center.

Sparck Jones, K., & Staveley, M.S. (1999). Phrasier: a system for interactive document retrieval using keyphrases. In Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'99), (pp. 160–167).

Spink, A., & Losee, R.M. (1996). Feedback in information retrieval. Annual Review of Information Science Technology, 31, 33–78.

van Riemsdijk, Henk, & Williams, E. (1979). Information retrieval. Cambridge, MA: MIT Press.

Voorhees, E.M., & Harman, D.K. (1998). Overview of the sixth text retrieval conference TREC-6. In Proceedings of the Sixth Text Retrieval Conference (TREC-6), NIST Special Publication 500240, 1–24.

Wen, J.-R. et al. (2001). Clustering user queries of a search engine. In Proceedings of the 10th International World Wide Web Conference (WWW'01), (pp. 162–168).

Worona, S. (1971). Query clustering in a large document space. In G. Salton (ed.), The SMART retrieval system: experiments in automatic document processing. Englewood Cliffs, NJ: Prentice Hall, 298–310.

Xu, J., & Croft, W.B. (1996). Query expansion using local and global document analysis. In Proceedings of 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR'96), (pp. 4–11).