RC24205 (W0703-018) March 6, 2007
Computer Science

IBM Research Report

MedSearch: A Specialized Search Engine for Medical Information

Gang Luo, Chunqiang Tang, Hao Yang
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Xing Wei
University of Massachusetts at Amherst

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

MedSearch: A Specialized Search Engine for Medical Information

Gang Luo[1] Chunqiang Tang[1] Hao Yang[1] Xing Wei[2]
IBM T.J. Watson Research Center[1] University of Massachusetts – Amherst[2]
{luog, ctang, haoyang}@us.ibm.com xwei@cs.umass.edu

ABSTRACT

In this paper, we present *MedSearch*, a specialized Web search engine that can assist ordinary Internet users to search for medical information. Today, people of different ages are thirsty for medical information to address their various healthcare needs. Existing Web search engines, however, cannot handle medical search well for several reasons. First, a medical information searcher is often uncertain about his exact questions, especially in the early stage of symptom development. Thus, he prefers to receive comprehensive, relevant information from the search results. However, existing Web search engines are optimized for precision by concentrating their search results on only a few topics. Second, an ordinary user is usually unfamiliar with medical terminology and has little medical background. A natural way for him to use a medical Web search engine is to pose long queries that describe his detailed symptoms in plain English – a way similar to what he would talk to a doctor. However, all existing Web search engines impose certain limits on query length. To overcome these obstacles, MedSearch uses several key techniques to improve its usability and the quality of search results. First, it accepts queries of extended length and reforms long queries into shorter queries by extracting a subset of important and representative words. This not only significantly increases the query processing speed but also improves the quality of search results. Second, it provides diversified search results based on information extracted offline from a large medical document set crawled from high-quality Web sites. Lastly, it suggests related medical phrases, extracted from the popular MeSH ontology, to help the user quickly digest search results and refine the query. We have evaluated MedSearch using medical questions posted on medical discussion forums. The results show that MedSearch can handle various medical queries effectively and efficiently.

1. INTRODUCTION

Health care is a major business in many countries. As has been reported in [29], 16% of the gross domestic product (GDP) of the United States came from the health care sector in year 2004. The statistics for many other countries are similar: 10.9% in Switzerland, 10.7% in Germany, 9.7% in Canada, and 9.5% in France. As the baby boomer generation reaches their retirement age and health care becomes more expensive, the percentage of GDP spent on health care will continue to increase.

A large part of health care is related to the management and retrieval of medical information. The widespread use of the Web has radically changed the way people acquire medical information. As mentioned in [36], every day, more Americans (6% of Internet users on an average day) search for medical information on the Web than visiting doctors. Doctors themselves are increasingly using Web search engines to facilitate diagnosis because of the difficulty in keeping up with the rapid development of medical knowledge [13]. [21] reported that 79% of Internet users have searched for medical information on the Web. Most of these users thought they obtained useful information online, and were more willing to use Web search engines rather than going to a particular health-related Web site. Half of these users said that they would resort to the Web first for their next health question. While not all the information on the Web is valid, most doctors and patients believe that access to such online resources is beneficial. This by no ways implies that the Web will replace doctors someday. Instead, people use Web resources to better prepare for doctors' appointments and to better digest information obtained from doctors afterwards. Due to the increasing lack of new doctors and the retirement of baby-boomer doctors [8], the interaction time between doctors and patients keeps shrinking, and this trend is expected to last in the foreseeable near future.

In response to this huge market need, Healthline [16], a popular Web search engine for medical information, came into existence in October 2005. Shortly thereafter, Google announced its own medical Web search engine, Google Health [14], in May 2006. There are also several other medical Web search engines [35]. While these systems have their own merits, they all treat medical search in much the same way as traditional Web search.

Medical search has several unique requirements that distinguish itself from traditional Web search. A common scenario in which a person performs medical search is that he feels uncomfortable but is uncertain about his exact medical problems. In this case, the searcher usually prefers to learn all kinds of knowledge that is related to his situation. However, existing medical Web search engines are optimized for precision and concentrate their search results on a few topics. This lack-of-diversity problem is aggravated by the nature of medical Web pages. When discussing a medical topic, many medical Web sites use similar, but not identical, descriptions by paraphrasing contents in medical textbooks and research papers. Hence, search results provided by existing medical Web search engines often contain much semantic redundancy, which cannot be easily handled by existing methods for identifying near-duplicate documents [7] or result diversification [11, 47, 48]. To find useful medical information, the searcher often has to go through a large number of Web pages laboriously.

Another unique feature of medical search is due to the fact that most Internet users do not have much medical knowledge. A medical information searcher is often unclear about the problem that he is facing and unaware of the related medical terminology (e.g., panophthalmitis). As a result, it is difficult for him to choose a few accurate medical phrases as a starting point for his search.

Instead, considering the importance of his health, the Web searcher is typically willing to take his time to describe his situation in great detail (e.g., his medical history, his family medical history, where and how he feels uncomfortable, and what happened in the last several days) by posing long queries in plain English [20, 23, 39], much like the way he talks to a doctor. Actually, a recent study on medical queries [40] has reported that medical information searchers (1) tend to solicit specific medical information by posing detailed queries, and (2) feel most convenient to formulate searches as readable and understandable queries. These queries often contain many words "useless" to Web search engines, because ordinary searchers do not understand how a Web search engine works [40]. However, putting aside such "useless" words, these queries still contain a large number of "useful" words due to the complicated nature of medical treatment. This can be illustrated by an analogy to the medical expert system, where the user needs to answer more than twenty questions in order to describe his situation in sufficient detail [12, 43]. If answers to these questions are transformed into a query *Q*, *Q* would easily contain 50-100 words. This is also consistent with our observation that many medical questions posted on medical forums contain several hundred words. Figure 1 shows one example of such queries.

> http://www.medhelp.org/forums/RespiratoryDisorders/messages/2584.html
> ... My 23 month old son has been coughing since 6 months old … Seems to be constantly on antibiotics for every kind of chest infection, on pulmicort, albuterol 2x's a day, constant ear infections (tubes, adnoids, and tonsils are scheduled), chronic loose stools. Seen an allergist, he has lots of environmental allergies, did all the mattres covers, rugs are gone, air purifier in. All this to no avail. Chest xray showed streaking in the main bronch tubes (?) perihilar stuff hazy areas, left lobe is alot grayer than the right. … Went to pedi pulmonologist in Boston, scheduled for sweat test on Friday, he doesnt think he has it, but wants to rule out CF. He wants to do CT and bronchoscope next week. Mentioned something about poss. deformed broch tubes, or weak lung walls, or even a cyst compressing his lungs causing this cough … what are the possibilities he has a verison of pulmonary micobacterial infection? ...

Figure 1. An exemplary medical question posted on the Med Help International Medical and Health Forum (http://www.medhelp.org/forums.htm).

Even after stopword removal, the above query still cannot be fed directly into existing medical Web search engines, because they all impose certain limits on query length for various reasons. For instance, the limits for Google and Healthline are 32 words [33] and 20 words [16], respectively. Google truncates long queries whereas Healthline simply rejects long queries. Such a low limit on query length is a serious obstacle for medical information searchers. Moreover, a medical information searcher often prefers the search engine to automatically suggest diversified, related medical phrases [4, 35, 45] that can help him quickly digest search results and refine his query. However, this cannot be done with existing medical Web search engines when the query is written using plain English description and has a terminological discrepancy from medical phrases.

In this paper, we present MedSearch, a prototype medical Web search engine that addresses the aforementioned limitations of existing systems. MedSearch uses several key techniques that significantly improve its usability and the quality of search results. First, MedSearch accepts queries of extended length and supports the use of plain English description. This is a great convenience for the majority of Internet users who do not have much medical knowledge. MedSearch automatically rewrites long queries into moderate-length queries by selectively dropping unimportant terms (i.e., words). Since unimportant terms not only appear in a large number of Web pages but also obscure the main theme of the query, dropping them can both significantly increase the query processing speed and improve the quality of search results [20]. Second, MedSearch returns diversified Web pages without significantly increasing query processing time or deteriorating the quality of the returned top Web pages, which allows the searcher to see various aspects related to his situation. Third, MedSearch automatically suggests diversified medical phrases, ordered by their relevance to the query, to the searcher. These medical phrases are extracted and ranked based on multiple sources: the standard MeSH [28] medical ontology, the collection of crawled Web pages, and the query itself.

There are several key challenges in designing MedSearch. In order to rewrite long queries into moderate-length queries, we must aggressively drop unimportant terms yet avoid losing much useful information. In providing diversified search results, one major challenge is to efficiently handle the excessive redundancy among different medical Web pages. When ranking the suggested medical phrases, we need to resolve the terminological discrepancy between medical phrases and queries written in plain English. For this purpose, a set of representative Web pages are computed offline for each medical phrase. (Note that the number of medical phrases is limited and does not grow with the corpus size.) Since a large part of these high-quality representative Web pages are written in plain English, they provide good linkages between medical terminology and plain English words. The relevance between a query *Q* and a medical phrase *M* is computed as a function of the relevance scores between *Q* and *M*'s representative Web pages. Then all the suggested medical phrases are sorted in descending order of their relevance scores.

With the capability of searching both relevant Web pages and related medical phrases, MedSearch can assist a patient throughout his entire medical treatment process:

(1) The patient can use MedSearch to facilitate preliminary self-diagnosis.

(2) The patient can use MedSearch to better prepare for doctor's appointments.

(3) During the appointment, the doctor may not explain everything in great detail. After coming back home, the patient can use MedSearch to help him digest the information that he does not fully understand.

(4) If the patient's situation is puzzling, even the doctor may not be able to provide a satisfactory solution. In this case, the patient can use MedSearch to find more information and clarify his symptom description. For example, it has been reported (http://www.webmd.com/content/pages/12/50275) that a patient could not get her ear problem solved for seven years because of miscommunication between her and the doctor. Later, she found through Web search a medical phrase (pulsatile tinnitus) that accurately describes her symptom, and then consulted the right specialist who immediately cured her ear problem.

We crawled a large number of medical Web pages from the Internet and evaluated the effectiveness of our techniques using medical questions that people posted on a medical forum. Our results show that MedSearch can process long queries efficiently, at a speed roughly comparable to that of existing medical Web search engines in processing short queries. Our experiments also show that user satisfaction is crucially tied to MedSearch's capability of returning diversified Web pages and suggesting diversified, related medical phrases that help users quickly understand the returned pages and refine their queries.

The rest of the paper is organized as follows. Section 2 provides some background on information retrieval. Section 3 presents the details of our techniques. Section 4 evaluates the effectiveness of our techniques under a wide variety of query scenarios. We conclude in Section 5.

2. BACKGROUND ON OKAPI

In this section, we review Okapi, a state-of-the-art method for ranking documents. In Section 3, we will show how MedSearch extends this method to work for medical search.

In Okapi [34, 37], both documents and queries are represented as vectors. Each element of a vector is the weight of a term in the vocabulary (i.e., the set of all the distinct words). Terms that are important to a document are assigned large weights. Terms that do not appear in the document have weights zero. The relevance between a document *D* and a query *Q* is computed as the inner product of *D*'s vector and *Q*'s vector.

The intuition behind Okapi is that the more times a term *t* appears in a document *D* and the fewer times *t* appears in other documents (i.e., the less popular *t* is in other documents), the more important *t* is for *D*. Also, the effect that longer documents have more words needs to be compensated by normalizing for document lengths.

Consider a query *Q* and a collection of documents *C*. For each term *t* in the vocabulary and a document $D \in C$, Okapi uses the following formulas:

(f1) term frequency (tf) weight

$$w_{tf} = (k_1 + 1)tf / \{k_1[(1-b) + b \times dl / avdl] + tf\},$$

(f2) inverse document frequency (idf) weight

$$w_{idf} = \ln[(N - df + 0.5)/(df + 0.5)],$$

(f3) query term frequency weight

$$w_{qtf} = (k_3 + 1)qtf / (k_3 + qtf),$$

(f4) term weight $w_t = w_{tf} \times w_{idf} \times w_{qtf}$,

(f5) $score_{D,Q} = \sum_{t \in D,Q} w_t$.

Here *tf* is *t*'s frequency (i.e., number of occurrences) in *D*, *qtf* is *t*'s frequency in *Q*, *N* is the total number of documents in *C*, *df* is the number of documents in *C* that contain *t*, *dl* is the length of *D* in bytes, and *avdl* is the average length (in bytes) of all the documents in *C*. *b*, k_1, and k_3 are three predetermined constants. Typically, as suggested in [37], *b=0.75*, $k_1 = 1.2$, and $1 \le k_3 \le 1000$. As described in Section 3.3 below, our MedSearch chooses $k_3 = 1$.

For each document $D \in C$, Okapi defines its relevance score (i.e., the degree of relevance for answering query *Q*) as in equation f5. This score is the sum of term weights of all the terms that appear in both *D* and *Q*.

3. HANDLING MEDICAL QUERIES

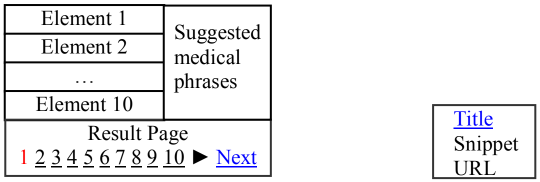
MedSearch is designed to assist ordinary Internet users who are unfamiliar with medical terminology and have limited medical background. Such users often are unclear about what they are looking for, especially during the early stage of medical treatment. Naturally they will pose long queries that describe their symptoms, medical history, etc., in detail using plain English. On the other hand, medical Web pages often are written by professionals and typically contain many medical jargons. The resulting gap between the medical terminology and the fuzzy queries in daily language presents a grand challenge for medical search.

To address this challenge, MedSearch makes use of the Medical Subject Headings (MeSH) ontology [28], a standard vocabulary edited by the National Library of Medicine and widely used for indexing and cataloging biomedical and health-related documents. The MeSH ontology is organized into a tree structure, whose branches correspond to different categories of medical phrases. MedSearch uses the information in the branches of the MeSH tree that correspond to categories A~G (i.e., anatomy, organism, diseases, chemicals and drugs, analytical, diagnostic and therapeutic techniques and equipment, psychiatry and psychology, biological sciences), as the other branches (e.g., humanities) do not contain the medical phrases that searchers care about. As we will see shortly, we use this ontology to identify medical phrases in the returned top Web pages and to rank medical phrases based on their relevance to the original query.

MedSearch crawls Web pages from a few selected, high-quality medical Web sites rather than all the Web sites. Such a vertical search engine [10] approach is also adopted by both Healthline [16] and Google Health [14], because a general-purpose search engine (e.g., Google) that collects pages from the entire Web can suffer from the disturbance of many low-quality pages in the search results [27].

3.1 User Interface

The user interface of MedSearch contains two parts: the query interface and the answer interface. In a traditional Web search engine, most input queries are short (e.g., containing less than ten words). Hence, the query interface, which accepts the input query from the searcher, is usually a single-line text field. In contrast, MedSearch accepts queries of extended length and uses a multi-line text area as the query interface [23, 38].



(a) high-level answer format (b) element format

Figure 2. The answer interface of MedSearch.

Figure 2 shows the format of MedSearch's answer interface. Similar to existing Web search engines, MedSearch organizes answers to a medical query into one or more result pages. Each result page contains ten elements. An element corresponds to a Web page *P* and contains the title, the snippet (i.e., some words extracted from *P*), and the URL of *P*. In addition, suggested medical phrases are listed on the right side of the result page. All these medical phrases belong to the MeSH ontology. Depending on the searcher's requirement, these medical phrases can be organized into different categories (e.g., diseases, treatments, drugs, organs) according to the classification in the MeSH ontology. When the searcher moves the mouse to a medical phrase *M*, the explanation of *M* that comes from the annotation field in the MeSH ontology is automatically displayed. This helps the searcher understand these suggested medical phrases.

3.2 Overview of Our Approach

Let *C* denote the collection of all the Web pages crawled by MedSearch. As standard pre-processing steps in Web information retrieval, for the Web pages in *C*, (1) all the HTML comments, JavaScript code, tags, and non-alphabetic characters are removed [17], (2) stopwords are removed by using the standard SMART stopword list [42], and (3) a forward index I_f and an inverted index I_i are built using the single-term vocabulary. In addition, another forward index I'_f that contains only medical phrases is built for the Web pages in *C*. MedSearch uses I'_f to suggest related medical phrases to the searcher.

MedSearch processes a medical query *Q* in the following steps:
Step 1: Remove stopwords from *Q*.
Step 2: Rewrite *Q* into a moderate length if it is too long.
Step 3: Produce search result pages.
Step 4: Generate snippets.
Step 5: Suggest related medical phrases.

3.3 Step 2: Rewriting Long Queries

As mentioned in the introduction, a medical query that uses plain English description can easily contain hundreds of terms even after stopword removal. In general, given a query *Q*, existing Web search engines use one of two methods to limit the number of Web pages that need to be considered in ranking Web pages:

(1) Only Web pages that contain all the terms in *Q* are considered. This is achieved by intersecting those inverted lists in the inverted index I_i that correspond to all the distinct terms in *Q*.

(2) All the Web pages that contain at least one term in *Q* are considered. This is achieved by computing the union of those inverted lists in I_i that correspond to all the distinct terms in *Q*.

Most major Web search engines use the first approach. Since almost none of the Web pages in the collection *C* contains all the terms in *Q*, this approach is unsuitable for long medical queries. Hence, MedSearch uses the second approach.

We notice that in a typical, long medical query *Q*, many of *Q*'s terms appear in a large number of Web pages in the collection *C* but do not carry much useful information. Especially, some terms can appear in 80%~90% of all the Web pages in *C*. As a result, if we use the second approach without modification, then almost all the Web pages in *C* need to be processed in answering a query, which is not scalable as the corpus size grows. Moreover, traversing the inverted lists in I_i that correspond to all the distinct terms in *Q* can be rather time-consuming. This problem is unique to medical search, as many medical queries are rather long and written in plain English. It does not appear in short keyword queries typically used in traditional Web search, where most query keywords carry content-related information and only appear in a small fraction of all the Web pages in *C*.

To avoid this problem, all existing medical Web search engines artificially impose rather restrictive limits on query length. This is particularly undesirable for medical search, as medical queries tend to be long due to their inherent fuzziness. An alternative solution is to ask the searcher to manually drop unimportant terms from his query. However, that is not only inconvenient to the searcher but also often impossible, as the importance of a term *t* depends on *t*'s distribution in the collection *C* that is unknown to the searcher. A more intelligent solution is for the search engine to automatically identify and drop unimportant terms from long queries so that the modified queries can be processed efficiently without sacrificing the quality of search results. This is the query rewriting method adopted in MedSearch.

The problem of handling (moderately) long queries has been studied before [20, 39]. The general approach is to replace the original query *Q* with its sub-queries that contain only a subset (e.g., three or four) of *Q*'s terms. [20] proposed generating a few "good" sub-query candidates by computing the mutual information scores of all possible sub-queries of *Q*, and then letting the user choose the final sub-query that is used to replace *Q*. This method has two limitations: (1) it is prohibitive to enumerate all possible sub-queries of long queries as those used in medical search, and (2) short sub-queries cannot fully represent the meanings of long queries. In contrast, [39] proposed using term weighting to form short sub-queries from *Q*, where the number of sub-queries increases super-linearly with the length of *Q*. These short sub-queries are sent to the Web search engine, and their retrieval results are merged to form the final result. Again, (1) the method in [39] is prohibitive for long medical queries because it submits many sub-queries to the Web search engine, and (2) short sub-queries cannot fully represent the meanings of long queries.

Next, we describe our query rewriting method in detail. Consider a medical query *Q* that contains $||Q||$ distinct terms. MedSearch uses a length threshold l_T to differentiate short queries from long queries. If $\|Q\| < l_T$, MedSearch treats *Q* as a short query and does not change *Q*. Otherwise, MedSearch treats *Q* as a long query and automatically rewrites *Q* into a moderate-length query Q' by selectively dropping unimportant terms. In our current implementation of MedSearch, the default value of l_T is *10*.

For all the terms in *Q*, their tf×idf values roughly reflect their importance (see Section 2). These tf×idf values are computed using the Okapi formula [34, 37] that is reviewed in Section 2: $w_{t,Q} = w_{qtf} \times w_{idf}$. Then all the terms in *Q* are sorted in descending order of their tf×idf values. Those terms that are ranked low are the candidates to be dropped from *Q*. In Equation (f2) that computes w_{qtf}, we set $k_3 = 1$. This reduces the influence of the query term frequency *qtf* on $w_{t,Q}$. Consequently, query terms with small idf values (i.e., those appearing in many Web pages) are less likely to have larger tf×idf values than query terms with large idf values. As mentioned before, keeping those query terms with small idf values in Q' not only slows down query processing but also deteriorates the quality of search results, as irrelevant terms obscure the main theme of the query.

To avoid overly long query processing time and improve the quality of search results, we set an upper bound *U* on the length of the modified query Q'. *U* is counted in the number of distinct terms. Only the top $m = \min(U, \|Q\| \times p)$ terms in *Q* with the largest tf×idf values are kept in Q', where *p* is a constant. For each term kept in Q', its number of occurrences in Q' is equal to that in *Q*. In our current implementation of MedSearch, *p=90%*. In practice, if *U* is too small, Q' cannot capture enough information in the original query *Q*. This will deteriorate the quality of search results. On the other hand, if U is too large, query processing can be rather slow. Also, the quality of search results will deteriorate due to the large number of irrelevant terms in Q'. Our experiments in Section 4.3 show that a good value for U is usually between 70 and 100. Note that the 32-word query length limit of Google counts both repeated terms and stopwords. After removing stopwords, the "effective" query length limit in Google that is counted as the number of distinct terms is much smaller than 32. Also, our method of dropping terms is more intelligent than the brute-force truncation method used in Google.

Typically, unimportant terms appear in a large fraction of the Web pages in the collection C while important terms appear in a smaller fraction of the Web pages in C. Hence, if the lowest-ranked $q\%$ of the terms in a query Q are dropped, typically we can reduce the number of Web pages that need to be processed for Q by much more than $q\%$. In other words, the query processing time is reduced by a factor much larger than $1/(1-q\%)$.

In traditional information retrieval, most queries are short and may not contain enough information for retrieving documents. To improve the quality of search results, relevance feedback or query expansion [3, 15] is used to add a limited number of relevant terms into the original query. In contrast, in medical search, the original query is often too long and contains many irrelevant terms that obscure the main theme of the query. In this case, dropping unimportant terms from the original query not only significantly reduces the query processing time but also improves the quality of search results.

The length upper bound U of modified queries affects the query processing speed. The larger the U, the more slowly queries are processed. When the system is heavily loaded, many Web search engines dynamically modify query execution to reduce the load [5, 26]. Similarly, our method can dynamically adjust U to control query processing time. The concrete method is as follows. The system administrator specifies three constants E, I, and T. If the average query processing time within the last I seconds is above E, we consider the system is overloaded. Let $[U_{min}, U_{max}]$ be the safe range of U specified by the system administrator. When U is within this range, the system administrator considers the quality of search results to be acceptable. The goal of our algorithm is to keep enough useful information in the modified queries without overloading the system. Initially, $U=U_{max}$. At any time, U is always kept within the range of $[U_{min}, U_{max}]$. Every T seconds, the system checks whether it is overloaded. If it is overloaded and $U>U_{min}$, U is decremented by one to reduce the system load. Otherwise if the system is not overloaded and $U<U_{max}$, U is incremented by one to increase the amount of useful information in the modified queries.

3.4 Step 3: Diversifying Search Results

MedSearch uses the Okapi method that is reviewed in Section 2 to rank Web pages. However, only using the Okapi method will concentrate the search results on a few topics. In the past, studies have shown that searchers usually prefer diversified search results [1, 11, 32, 47, 48]. The existing methods for result diversification fall into three categories:

(1) Re-rank or cluster the returned top-L Web pages [11, 18].

(2) Generate from the original query a set of related queries, and then use them to perform search [32].

(3) Rank all the Web pages according to a hybrid score that combines both a dynamically computed relevance score and a statically pre-computed diversity score [48].

These methods were initially developed for traditional Web search. They did not consider the following unique properties of medical search:

(1) As mentioned in the introduction, Web pages from medical Web sites are highly redundant. The returned top-L Web pages often cover only a few topics and sometimes cover only a single topic. Regardless of how clustering or re-ranking is performed, it is unlikely to find enough diversified search results from these top Web pages. This phenomenon has been observed elsewhere before [18]. This problem cannot be solved by simply increasing the number L of returned top pages, as it is difficult to determine a proper value of L for all queries. Also, when L is too large, online clustering or re-ranking can be rather expensive for an interactive medical Web search engine.

(2) The method in [32] learns related queries from query logs by analyzing queries' repetition pattern. Short queries may repeat, but not long queries. Hence, the method in [32] does not work for long queries that are frequently encountered in medical search.

(3) In [48], the diversity score is also called the affinity ranking (AR) score. The AR score of a Web page P_i is discounted by the weighted AR scores of all the Web pages P_j that are ranked before P_i according to certain information richness criterion, where the weights $\widetilde{M}_{i,j}$ are the normalized similarity scores between P_i and P_j's ($\sum_j \widetilde{M}_{i,j}=1$). As mentioned before, Web pages from medical Web sites are highly redundant. Consider a topic that is repeatedly mentioned by a large set S of similar Web pages. For each Web page $D_i \in S$, its $\widetilde{M}_{i,j}$'s are small, as they are normalized by the contributions from the large number of Web pages in S. Consequently, the top ranked Web pages (determined by the information richness criterion) in S will have similar AR scores, and the result diversification method in [48] cannot work well for these Web pages.

To address the limitations mentioned above, MedSearch uses a novel pre-clustering method to provide diversified search results for medical queries. Our method does most computation offline, and has minimal negative impacts on online query processing speed and the quality of the returned top few Web pages. In a pre-processing step, all the Web pages in the collection C are clustered into K clusters. Each of these K clusters roughly corresponds to a different topic. For each Web page in C, its cluster number is recorded in the forward index I_f and can be easily retrieved. The system administrator specifies a constant J ($J<K$ and $J=20$ by default) that controls the diversity of search results. When ranking Web pages, each cluster can contribute at most one Web page to the returned top-J Web pages. In other words, all the returned top-J Web pages belong to different clusters and are sorted in descending order of their relevance scores. This is done by recording the Web page with the highest relevance score for each of the K clusters. Starting from the $(J+1)$th page, the remaining returned Web pages are ranked in the usual way, i.e., in descending order of their relevance scores. Using this method, the searcher is likely to see different aspects in the returned top-J Web pages. Moreover, many of these J Web pages are likely to be relevant to the query, as these J Web pages have the highest relevance scores in the corresponding clusters.

There is one exception in the above description. Suppose that all the Web pages that are under consideration for the query Q

(see Section 3.3) belong to K' clusters, where $K' \le K$. If $K' < J$, it is impossible for all the returned top-J Web pages to belong to different clusters. In this case, we require that all the returned top-K' Web pages belong to different clusters.

MedSearch uses the K-means algorithm [41] to perform pre-clustering, as K-means is one of the most robust methods for document clustering. How to estimate the optimal value of K and how to update the clusters to handle continuously arriving documents are orthogonal to our search result diversification method, and there are some known solutions [9, 31, 46]. Nevertheless, we observed in our experiments (in Section 4.3) that the performance of our system is not sensitive to the value of K as long as K is within a reasonable range.

3.5 Step 4: Generating Snippets

After obtaining the search result Web pages, MedSearch uses the standard passage retrieval technique [22, 24] to generate a snippet for each page. For each such snippet s_n, MedSearch highlights in s_n the medical phrases and the top-3 common terms between s_n and the query Q that have the largest tf×idf values in Q.

3.6 Step 5: Suggesting Related Medical Phrases

One unique issue in medical search is that searchers are typically unfamiliar with medical terminology (e.g., panophthalmitis). Therefore, reading the returned Web pages can be difficult and time-consuming, especially when the searcher needs to refine his query multiple times before he eventually finds the desired information. During such an iterative search process, the quality of search results can be gradually improved by adding accurate medical phrases into the query. However, this is difficult to do for most searchers due to lack of medical knowledge.

To solve these problems, Healthline [16] automatically suggests related medical phrases to the searcher based on his query. (But Healthline does not provide any explanation of these suggested medical phrases as what MedSearch does.) From the searcher's perspective, scanning these suggested medical phrases is much faster than reading the returned Web pages, and can quickly help query refinement. As a result, this feature of Healthline is highly attractive to medical information searchers [4].

However, the method that Healthline uses to suggest related medical phrases has several limitations. All the suggested, related medical phrases come from a medical taxonomy that is manually edited by 1,100 doctors over several years. For a given query, Healthline suggests related medical phrases according to certain rules. In neither the construction of the taxonomy nor the process of suggesting related medical phrases does Healthline perform any statistical analysis on the query or the crawled Web pages [4]. Obviously, this method is extremely labor-intensive and has limited scalability. Moreover, Healthline rejects queries that contain more than 20 words and does not suggest any related medical phrase for them.

For short medical queries, [45] proposes a method that maps a query Q into one or more medical phrases M with the smallest editing distance from Q, and then recommends medical phrases that are "semantically" close to M. This method is problematic for long medical queries because of the difficulty of mapping a long query into medical phrases solely based on editing distance.

To overcome the limitations of existing methods, MedSearch uses a statistical method to suggest related medical phrases, by analyzing medical phrases in the MeSH ontology, the crawled Web pages, and the query. For each query, MedSearch suggests V related medical phrases, where V is a constant specified by the system administrator. To ensure a high probability that some of these V medical phrases are desired by the searcher, V should not be too small. On the other hand, to avoid overwhelming the searcher and to fit the V medical phrases into the right side of the answer interface (see Figure 2), V should not be too large either. The default value of V in MedSearch is 60.

The suggestion process consists of two sub-steps. The first sub-step is to generate the candidate set S of related medical phrases. The second sub-step is to rank the medical phrases in S. A main challenge in the second sub-step is due to the fact that medical phrases use medical terminology while the query uses plain English description. Resolving this terminological discrepancy is crucial to providing an appropriate ranking of the suggested medical phrases. Next, we describe these two sub-steps in detail.

In the first sub-step, MedSearch selects V medical phrases from the returned top-J Web pages, where J is defined in Section 3.4. As mentioned in [30, 49], the suggested medical phrases need to be both relevant and diverse in order to provide the greatest convenience to the searcher. Intuitively, to ensure that a medical phrase M is relevant, it is better for M to appear in one of the returned top Web pages with a large tf×idf value. To ensure enough diversity in the list of suggested medical phrases, a single Web page should not contribute too many medical phrases to that list. We use a continuous discounting method to achieve these two goals. Each time a medical phrase is selected from a Web page P, a discount is given to the tf×idf values of the remaining medical phrases in P. As a result, the more medical phrases have already been selected from P, the less likely the remaining medical phrases in P will be selected in the future. The concrete method is as follows.

For each of the returned top-J Web pages, we find all its medical phrases and compute their tf×idf values using the Okapi formula that is reviewed in Section 2: $w_{t,D} = w_{tf} \times w_{idf}$. In this process, we do not consider the medical phrases in the query, as the searcher already knows them. We obtain a list L_t of triplets (medical phrase M, Web page P, tf×idf value $w_{M,P}$), and select V distinct medical phrases from L_t to form a candidate set S. This is done in V passes. In each pass, a medical phrase M' with the largest tf×idf value is selected from L_t. Then all the triplets with the same medical phrase M' are dropped from L_t, as we are only interested in distinct medical phrases. For all the remaining medical phrases in the Web page where M' comes from, their tf×idf values are discounted by a factor d that is specified by the system administrator. The default value of d in MedSearch is 0.9.

The above algorithm is described in pseudo code as follows:

```
Let set $S = \phi$.
For ($i = 1$ ; $i \le V$ ; $i++$) {
    Scan all the triplets in $L_t$ and find the triplet $t'_r = (M', P', w_{M',P'})$ with the largest tf×idf value.
    Let $S = S \cup \{M'\}$.
    Drop all the triplets from $L_t$ whose $M = M'$.
    For all the triplets $t_r = (M, P, w_{M,P})$ in $L_t$ whose $P = P'$,
    let its $w_{M,P} = w_{M,P} \times d$.
}
```

Note that if the returned top-*J* Web pages contain V' distinct medical phrases and $V' < V$, we can only obtain V'(rather than *V*) medical phrases from these Web pages. Moreover, after a discount has been given to the triplet $(M, P, w_{M,P})$ several times (i.e., the Web page *P* has already contributed several medical phrases to the candidate set *S*), it will become difficult for the medical phrase *M* to come out from *P* in the future. However, if *M* exists in some other Web page P'' and no (or not much) discount has been given to P'', *M* may still be able to come out from P''in the future.

In the second sub-step, we rank all the medical phrases in the candidate set *S* and present them to the searcher. A simple method, which we call the *tf×idf method*, is to rank all these medical phrases in the order that they are generated in the first sub-step. As we will show in Section 4.4, the quality of the resulting order is often unsatisfactory. This is because in a Web page *P*, those medical phrases with the largest tf×idf values may not be relevant to the query *Q*. For example, *P* has several aspects. One aspect is related to *Q* but the medical phrases in *P* with the largest tf×idf values describe the other aspects. A better method, which we call the *relevance score method*, is to rank all these medical phrases in descending order of their relevance scores for *Q*. Intuitively, this method is reasonable but it cannot be implemented in a straightforward way due to terminological discrepancy. We cannot directly compute the relevance scores between *Q* and the medical phrases in the candidate set *S*, because some medical phrases in *S* are relevant to *Q* but simply do not appear in *Q*.

In general, there are two alternatives to address this terminological discrepancy problem: (1) "translating" *Q* into medical terminology, or (2) "translating" medical phrases into plain English description. We find that the second approach is more practical and adopt it in MedSearch. Our basic idea is to convert each medical phrase $M \in S$ into *r* representative Web pages, where *r* is a constant. Many sections in the Web pages are written using plain English description, which matches with the language of the query *Q*. We compute the relevance score between *M* and *Q* as a weighted average of the relevance scores between *Q* and *M*'s representative Web pages. Unlike existing method [25] that selects topic words to summarize the returned top documents, the purpose of our algorithm is to give high ranks to the most relevant medical phrases. This helps the searcher refine his query.

There are several ways to select the representative Web pages for each medical phrase *M* in the MeSH ontology:

(1) We can ask medical experts to either manually select or specifically write *r* representative Web pages for *M*. In this case, we would expect these representative Web pages to be of high quality. However, this method is labor intensive.

(2) If the quality of the Web page collection *C* is good, we can use *M* as the query and find the top-ranked *r* Web pages R_i ($1 \le i \le r$) in *C*. These Web pages then serve as the representatives of *M*. In this case, we would expect these representative Web pages to have reasonably good quality.

(3) If the overall quality of the collection *C* is limited (e.g., including spams), we cannot expect all the retrieved R_is' to have good quality. In this case, an extra step is needed to improve the quality of representative Web pages.

For MedSearch, our situation falls into the second case, as MedSearch is a vertical search engine that crawls Web pages from a few selected, high-quality Web sites. For each medical phrase *M* in the MeSH ontology, we retrieve the top-ranked *r* Web pages in *C* and use them as *M*'s representative Web pages. These Web pages are recorded in a data structure and can be easily retrieved. This procedure is done offline and does not affect the online query processing time.

The relevance score between the query *Q* and a medical phrase $M \in S$ is computed as a weighted average of the relevance scores between *Q* and *M*'s representative Web pages:

$$score_{M,Q} = \sum_{i=1}^{r} score_{R_i,Q} / i \, .$$

Here, the weight for the *i*-th ($1 \le i \le r$) representative Web page R_i is $1/i$, and R_i's relevance score is computed using the Okapi method. Then all the medical phrases in *S* are sorted in descending order of their relevance scores for *Q*.

4. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of our proposed techniques, we conducted experiments using medical questions that people posted on a medical discussion forum.

4.1 Setup

We crawled 20GB of Web pages from WebMD [44], one of the most popular medical Web sites, and fed MedSearch with natural medical queries we extracted from a well-known medical forum. Such posts on medical forums might have different structures from medical queries that users would send to a Web search engine. However, we emphasize that both of them share the same key features, such as long queries, plain English description, and lack of accurate medical phrases, which are important in evaluating the performance of our system. Moreover, there is currently no trace of long medical queries as they cannot be accepted by existing Web search engines. As our prototype system obtains more users in the future, we plan to re-evaluate our system using real query traces once they become available.

We selected 30 representative questions that people posted on a popular medical forum, the Med Help International Medical and Health Forum (http://www.medhelp.org/forums.htm). These 30 queries cover a broad range of medical topics, including arthritis, respiratory disorder, gastric disorder, neurological disorder, cardiological disorder, eye disorder, dermatologic disorder, ovarian cancer, family practice, and menopause. One such query was shown earlier in Figure 1 in the Introduction.

Both relevance and diversity are judged using a single metric: *usefulness*. A returned Web page *P* is useful if *P* is relevant to the query, and much of *P*'s relevant content has not been mentioned in the Web pages that are ranked higher. If *P* is useful, its usefulness score $score_u(P) = 1$; otherwise, $score_u(P) = 0$. A similar definition of usefulness holds for the suggested medical phrases.

For the returned top-20 Web pages P_i ($1 \le i \le 20$), their weighted average usefulness score is defined as

$$avg\_score_u = \sum_{i=1}^{20} score_u(P_i) / \log(1+i) \, .$$

This is the NDCG metric used in [2, 19] for judging the quality of Web search results when there are two integer relevance labels (0 and 1). For the suggested *V=60* medical phrases, their weighted average usefulness score is defined similarly. The mean of the weighted average usefulness score over the 30 queries is the main quality metric for the returned Web pages and the suggested medical phrases.

Five colleagues served as assessors and independently determined the usefulness scores of the returned Web pages and the suggested medical phrases. None of them has formal medical training. The default parameter values used in our techniques are as follows: *U=80* (the length upper bound of the modified query), *K=1,500* (the number of clusters used in the pre-clustering method), and *r=1* (the number of representative Web pages for each medical phrase). Our experiments were performed on a computer with one 1.6GHz processor, 1GB memory, one 75GB disk, and running the Microsoft Windows XP operating system.

4.2 An Example

To give the reader a feeling of the contents returned by MedSearch, we present detailed results of the returned Web pages and the suggested medical phrases for the particular query in Figure 1. Table 1 shows the returned relevant Web pages. Table 2 shows the suggested relevant medical phrases. In general, for a medical query *Q*, MedSearch can find several relevant Web pages and medical phrases that cover multiple aspects of *Q*. These Web pages and medical phrases can be related to various topics, such as diseases, tests, examinations, drugs, and organs.

Table 1. Returned relevant Web pages.

rank	URL	topic
1	http://www.webmd.com/content/chat_transcripts/1/108027.htm?printing=true	asthma
3	http://www.webmd.com/hw/ear_disorders/hw184529.asp@printing=true	ear infection
4	http://www.webmd.com/content/chat_transcripts/1/107597.htm?printing=true	spring allergies
6	http://www.webmd.com/hw/lung_disease/hw32162.asp@printing=true	acute bronchitis
7	http://www.webmd.com/hw/lab_tests/hw5693.asp@printing=true	sputum culture
8	http://www.webmd.com/content/article/105/107786.htm?printing=true	sarcoidosis
12	http://www.webmd.com/hw/infection/hw207304.asp@printing=true	tuberculosis
13	http://www.webmd.com/hw/pneumonia/hw63870.asp@printing=true	pneumonia
14	http://www.webmd.com/hw/cold_and_flu/hw85335.asp@printing=true	cough
16	http://www.webmd.com/hw/lung_disease/aa33397.asp@printing=true?printing=true	chronic obstructive pulmonary disease

Table 2. Suggested relevant medical phrases.

rank	medical phrase
1	bronchoscopy
2	bronchitis
4	sarcoidosis
9	pneumonia
16	otitis media with effusion
17	severe acute respiratory syndrome
27	lymph
29	adenoids
45	albuterol
46	mycobacterium tuberculosis

4.3 Sensitivity Analysis of Parameter Values

There are several important parameters used in our techniques. In this section, we evaluate the impact of parameter values on the quality of search results (i.e., returned Web pages and suggested medical phrases) and query processing time by a set of experiments. In each experiment, we varied the value of one parameter while keeping the other parameters fixed.

The first experiment concerns *U*, the length upper bound of the modified query (see query rewriting in Section 3.3). The default value of *U* is 80. We varied *U* from 20 to 120. For the returned top-20 Web pages and the suggested 60 medical phrases, Figure 3 and Figure 4 show the impact of *U* on the weighted average usefulness score, respectively. (Note that, to make figures in Sections 4.3 and 4.4 more readable, the y-axis does not always start from zero.) In general, when *U* is too small, not enough information is kept in the modified query, which deteriorates the quality of search results. When *U* is too large, many irrelevant terms are included in the query and obscure its main point, which also deteriorates the quality of search results. Our query rewriting method achieves the best quality of search results when *U* is between 70 and 100.

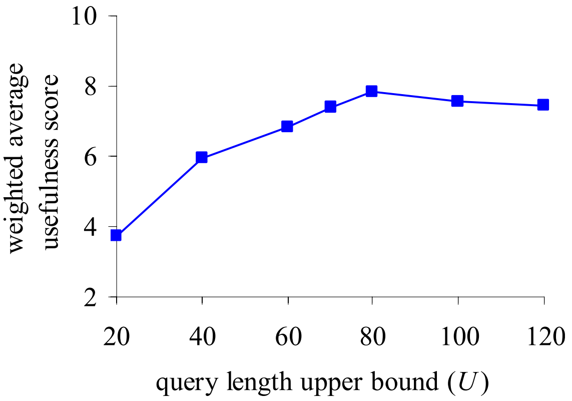


Figure 3. Weighted average usefulness score vs. *U* (Web page).

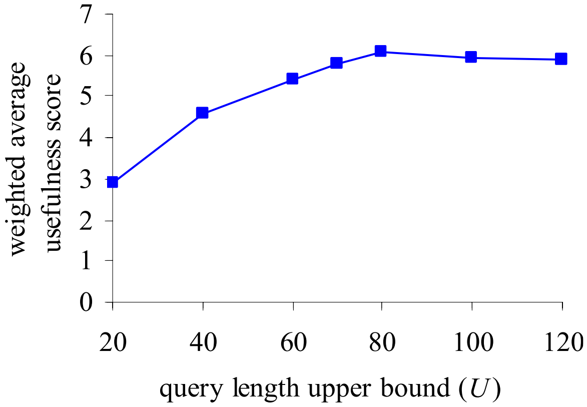


Figure 4. Weighted average usefulness score vs. *U* (medical phrase).

When *U*=80, the means of the weighted average usefulness scores for the returned top-20 Web pages and the suggested 60 medical phrases are 7.9 and 6.1, respectively. We present a simple calculation below to provide some intuition on these numbers. Let ws_i denote the weighted average usefulness score when the returned top-*i* Web pages (or medical phrases) are useful while the others are not useful. In this case, $ws_1 = 3.3$, $ws_2 = 5.4$, $ws_3 = 7.1$, and $ws_4 = 8.5$.

Figure 5 shows the impact of *U* on the average query processing time. The dotted horizontal line represents the average query processing time when the query rewriting method is not used (i.e., all the terms are kept in the query). As *U* increases, more terms are kept in the modified query *Q′* and it takes longer to process *Q′*. When *U*=80, the average query processing time is 0.6 second, which is 45% of the average query processing time when the query rewriting method is not used. As will be shown in Section 4.4, in this case, the weighted average usefulness score of the query rewriting method is higher than that when the query rewriting method is not used. Therefore, using an appropriate value of *U*, the query rewriting method simultaneously improves the quality of search results and reduces the query processing time.

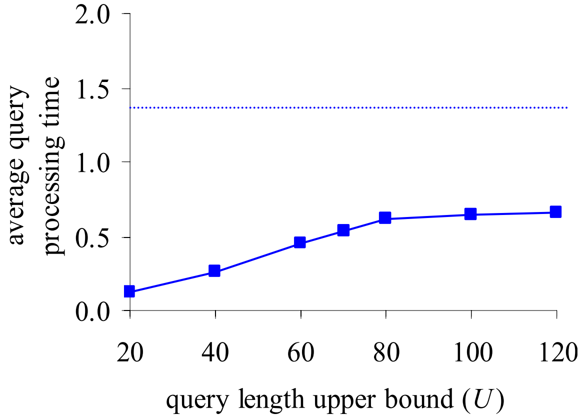


Figure 5. Average query processing time vs. *U*.

The second experiment concerns *K*, the number of clusters that is used in the pre-clustering method (see Section 3.4). The default value of *K* is 1,500. We varied *K* from 500 to 3,000. For the returned top-20 Web pages and the suggested 60 medical phrases, Figure 6 and Figure 7 show the impact of *K* on the weighted average usefulness score, respectively. In general, when *K* is too small, relevant Web pages tend to gather in the same clusters. Since each cluster contributes at most one Web page to the returned top-20 Web pages, we cannot find enough relevant search results from the top-20 clusters. When *K* is too large, the clustering effect is not significant and we cannot find enough search results that are both diversified and relevant. For the Web page collection used in our experiment, a good setting for *K* is between 1,000 and 2,000. As mentioned before, the problem of estimating the optimal value of *K* is orthogonal to our search result diversification method, and already has some known solution [31].

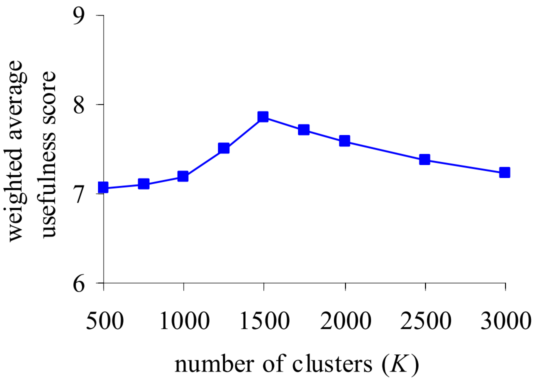


Figure 6. Weighted average usefulness score vs. *K* (Web page).

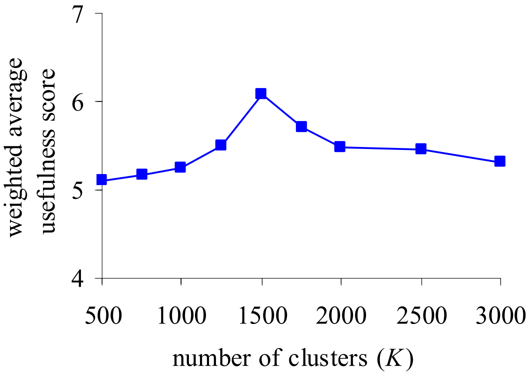


Figure 7. Weighted average usefulness score vs. *K* (medical phrase).

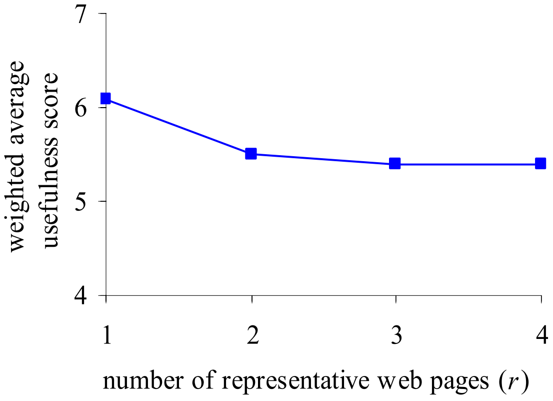


Figure 8. Weighted average usefulness score vs. *r* (medical phrase).

The third experiment concerns *r*, the number of representative Web pages for each medical phrase (see the relevance score method in Section 3.6). The default value of *r* is 1. We varied *r* from 1 to 4. For the suggested 60 medical phrases, Figure 8 shows the impact of *r* on the weighted average usefulness score. In general, for a medical phrase, the higher-ranked representative Web pages are more relevant than the lower-ranked representative Web pages. Hence, the weighted average usefulness score decreases as *r* increases. To achieve good performance, it is best to set *r=1*.

In summary, each of the MedSearch parameters has a not-very-small safe range that allows MedSearch to reliably achieve good performance. That is, the quality of search results is insensitive to parameter changes in this safe range. However, if a parameter value is outside its safe range, the quality of search results may degrade.

4.4 Influence of Individual Techniques

MedSearch adopts several key techniques that distinguish itself from existing medical Web search engines:

(1) **Technique 1**: Use the query rewriting method to rewrite long queries into a moderate length.

(2) **Technique 2**: Use the pre-clustering method to diversify search results.

(3) **Technique 3**: Use the relevance score method to rank the suggested medical phrases.

In this section, we evaluate the impact of individual techniques on the quality of search results using a set of experiments. In each experiment, we dropped one of the above three techniques while keeping the others intact. When Technique 1 is not used, all the terms are kept in the query. When Technique 2 is not used, no result diversification is performed. When Technique 3 is not used, the tf×idf method described in Section 3.6 (i.e., ranking all the medical phrases in the order that they are generated in the first sub-step) is used to rank the suggested medical phrases.

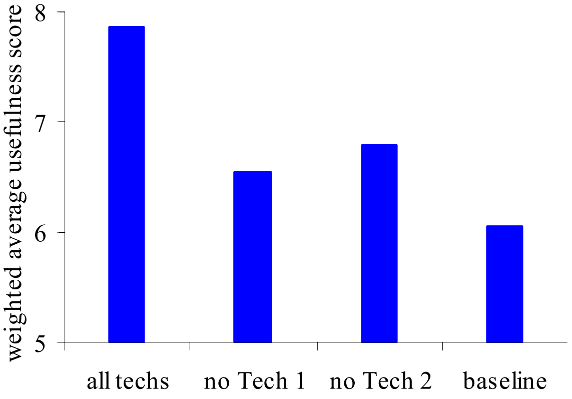


Figure 9. Weighted average usefulness score vs. used techniques (Web page).

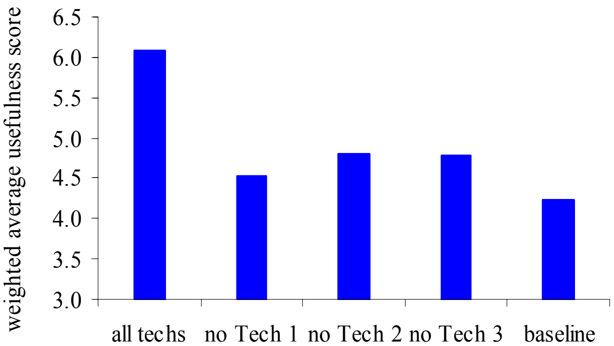


Figure 10. Weighted average usefulness score vs. used techniques (medical phrase).

For the returned top-20 Web pages and the suggested 60 medical phrases, Figure 9 and Figure 10 show the impact of the used techniques on the weighted average usefulness score, respectively. In both figures, "tech" stands for technique. "No Tech *i*" (*i*=1, 2, 3) represents the case that Technique *i* is not used. (Technique 3 has no impact on the returned Web pages and thus is not shown in Figure 9.) Baseline represents the case that none of the three techniques is used. The results clearly show that all the techniques used in MedSearch are necessary. If any of them is not used, the quality of search results degrades. Also, when all these techniques are used together, MedSearch performs much better than the baseline case: 30% improvement in the weighted average usefulness score for returned Web pages, and 44% improvement in the weighted average usefulness score for suggested medical phrases.

4.5 Comparison with Existing Search Engines

In this section, we compare MedSearch with two state-of-the-art medical Web search engines: Google Health [14] and Healthline [16]. Recall that the query length limits for Google Health and Healthline are 32 and 20 words, respectively [33, 16]. For each of the 30 queries, we used our query rewriting method in Section 3.3 to select the top *W* terms with the largest tf×idf values. These *W* terms were sent to both Google Health and Healthline as a modified query to accommodate the query length limits. We varied *W* from 5 to 20.

Google Health essentially only considers those Web pages that contain all the terms in the query [6]. Since almost no Web page contains all the *W* terms in the modified query, Google Health barely returns any result for the 30 queries. Healthline does not have this limitation. Figure 11 shows the weighted average usefulness score for the top-20 Web pages returned by Healthline. The dotted horizontal line represents the performance of MedSearch using the default parameter values. Since longer queries contain more useful information, the weighted average usefulness score of Healthline increases with *W*. MedSearch significantly outperforms Healthline, as Healthline does not perform result diversification and the *W*≤20 terms in the modified query of Healthline do not keep enough information (see Figure 3).

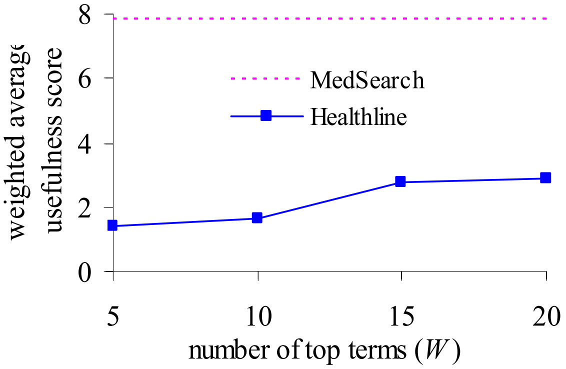


Figure 11. Weighted average usefulness score vs. *W* (Web pages returned by Healthline).

It is difficult to make a quantitative comparison between the related medical phrases suggested by Healthline and those suggested by MedSearch, as the output formats of these two systems are completely different. Healthline classifies the suggested medical phrases into several categories: broaden search, narrow search, and related topics. There is no global ordering among all the suggested medical phrases. For the 30 queries, Healthline often suggests very few (e.g., two) medical phrases. Even for the few queries that Healthline does suggest a reasonable number of medical phrases, those suggested phrases are highly redundant because Healthline does not perform result diversification.

5. CONCLUSION

This paper presents MedSearch, a specialized Web search engine for medical information retrieval. It can help ordinary Internet users throughout their entire process of medical treatment. The design of MedSearch takes into consideration the unique requirements of medical search. MedSearch supports queries written in plain English, accepts long queries, provides diversified search results, and suggests related medical phrases with proper ranking and annotation. These features are very attractive to the majority of Internet users who have little medical knowledge and are unfamiliar with medical terminology. Using medical questions that people posted on a medical forum, our experiments show that search result diversification and annotation significantly improve user satisfaction. In addition, MedSearch can process long queries at a speed comparable to that of traditional Web search engines in processing short queries.

6. REFERENCES

[1] A. Anagnostopoulos, A.Z. Broder, and D. Carmel. Sampling Search-Engine Results. WWW 2005: 245-256.

[2] E. Agichtein, E. Brill, and S.T. Dumais. Improving Web Search Ranking by Incorporating User Behavior Information. SIGIR 2006: 19-26.

[3] R.A. Baeza-Yates, B.A. Ribeiro-Neto. Modern Information Retrieval. ACM Press/Addison-Wesley, 1999.

[4] W. Boswell. Healthline.com - A Medical Search Engine. http://websearch.about.com/od/enginesanddirectories/a/healthline.htm.

[5] E.A. Brewer. Lessons from Giant-Scale Services. IEEE Internet Computing 5(4): 46-55, 2001.

[6] S. Brin, L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks 30(1-7): 107-117, 1998.

[7] A.Z. Broder. Identifying and Filtering Near-Duplicate Documents. CPM 2000: 1-10.

[8] A Future Doctor Shortage? A Growing Lack of New Doctors could be American's Next Health Crisis. http://www.cbsnews.com/stories/2006/06/17/eveningnews/main1726479.shtml, 2006.

[9] M. Charikar, C. Chekuri, and T. Feder et al. Incremental Clustering and Dynamic Information Retrieval. STOC 1997: 626-635.

[10] M. Chau, H. Chen. Comparison of Three Vertical Search Spiders. IEEE Computer 36(5): 56-62, 2003.

[11] J.G. Carbonell, J. Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. SIGIR 1998: 335-336.

[12] EasyDiagnosis medical expert system homepage. http://easydiagnosis.com.

[13] 'Googling' Aids Difficult Diagnoses. http://www.e-health-insider.com/news/item.cfm?ID=2258, 2006.

[14] Google Health homepage. http://www.google.com/Top/Health.

[15] D. Harman. Relevance Feedback Revisited. SIGIR 1992: 1-10.

[16] Healthline homepage. http://www.healthline.com.

[17] T.H. Haveliwala, A. Gionis, and D. Klein et al. Evaluating Strategies for Similarity Search on the Web. WWW 2002: 432-442.

[18] M.A. Hearst, J.O. Pedersen. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. SIGIR 1996: 76-84.

[19] K. Järvelin, J. Kekäläinen. IR Evaluation Methods for Retrieving Highly Relevant Documents. SIGIR 2000: 41-48.

[20] G. Kumaran, J. Allan. A Case for Shorter Queries, and Helping Users Create Them. HLT 2007.

[21] M. Klein, H. Easley. Checking Medical Facts Online can be OK, but don't Become a 'Cyberchondriac'. The Journal News, June 26, 2006. http://www.thejournalnews.com/apps/pbcs.dll/article?AID=/20060626/NEWS03/606260311/1019.

[22] C.C. Kwok, O. Etzioni, and D.S. Weld. Scaling Question Answering to the Web. TOIS 19(3): 242-262, 2001.

[23] R. Kraft, F. Maghoul, and C. Chang. Y!Q: Contextual Search at the Point of Inspiration. CIKM 2005: 816-823.

[24] M. Kaszkiel, J. Zobel. Passage Retrieval Revisited. SIGIR 1997: 178-185.

[25] D. Lawrie, B.W. Croft, and A.L. Rosenberg. Finding Topic Words for Hierarchical Summarization. SIGIR 2001: 349-357.

[26] X. Long, T. Suel. Optimized Query Execution in Large Search Engines with Global Page Ordering. VLDB 2003: 129-140.

[27] Medical Search Engine Rated 'Better Than Google'. http://www.ehiprimarycare.com/news/item.cfm?ID=2318, 2006.

[28] MeSH homepage. http://www.nlm.nih.gov/mesh/meshhome.html, 2006.

[29] The National Coalition on Health Care. Facts on the Cost of Health Care. http://www.nchc.org/facts/2006%20Fact%20Sheets/Cost%20-%202006.pdf, 2006.

[30] T. Nomoto, Y. Matsumoto. A New Approach to Unsupervised Text Summarization. SIGIR 2001: 26-34.

[31] D. Pelleg, A.W. Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. ICML 2000: 727-734.

[32] F. Radlinski, S. Dumais. Improving Personalized Web Search using Result Diversification. SIGIR 2006: 691-692.

[33] L. Rosenberger. Google Maximum Search Length Increased. http://lbr.library-blogs.net/google_maximum_search_length_increased.htm, 2005.

[34] S.E. Robertson, S. Walker, and M. Hancock-Beaulieu. Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive. TREC 1998: 199-210.

[35] SearchMedica - The GPs search engine. http://www.searchmedica.co.uk/searchmedica/EUIHomeAction.do, 2006.

[36] C. Sherman. Curing Medical Information Disorder. http://searchenginewatch.com/showPage.html?page=3556491, 2005.

[37] A. Singhal. Modern Information Retrieval: A Brief Overview. IEEE Data Eng. Bull. 24(4): 35-43, 2001.

[38] B. Shneiderman, D. Byrd, and W.B. Croft. Clarifying Search: A User-Interface Framework for Text Searches. D-Lib Magazine, January 1997.

[39] J. Shapiro, I. Taksa. Constructing Web Search Queries from the User's Information Need Expressed in a Natural Language. SAC 2003: 1157-1162.

[40] A. Spink, Y. Yang, and J. Jansen et al. A Study of Medical and Health Queries to Web Search Engines. Health Information and Libraries Journal 21(1): 44–51, 2004.

[41] M. Steinbach, G. Karypis, and V. Kumar. A Comparison of Document Clustering Techniques. Text Mining Workshop, KDD 2000.

[42] SMART Stopword List. http://www.lextek.com/manuals/onix/stopwords2.html, 2006.

[43] YourDiagnosis medical expert system homepage. http://www.yourdiagnosis.com.

[44] WebMD homepage. http://www.webmd.com.

[45] Q.T. Zeng, J. Crowell, and R.M. Plovnick et al. Assisting Consumer Health Information Retrieval with Query Recommendations. JAMIA 13(1): 80-90, 2006.

[46] O. Zamir, O. Etzioni. Web Document Clustering: A Feasibility Demonstration. SIGIR 1998: 46-54.

[47] C. Zhai, W.W. Cohen, and J.D. Lafferty. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. SIGIR 2003: 10-17.

[48] B. Zhang, H. Li, and Y. Liu et al. Improving Web Search Results Using Affinity Graph. SIGIR 2005: 504-511.

[49] C. Ziegler, S.M. McNee, and J.A. Konstan et al. Improving Recommendation Lists through Topic Diversification. WWW 2005: 22-32.