Textual Document Indexing and Retrieval via Knowledge Sources and Data Mining

Wesley. W. Chu, Zhenyu Liu and Wenlei Mao
Computer Science Department, University of California, Los Angeles 90095
{wwc, vicliu, wenlei}@cs.ucla.edu

Abstract We present a knowledge-based query expansion technique to improve document retrieval effectiveness. The general concept terms in a query are substituted by a set of specific concept terms used in the corpus that co-occur with the key query concept. Since the expanded query matches with the document index terms much better, experimental results reveal that such query expansion produces better retrieval effectiveness than the unexpanded ones.

We have also developed a new phrase-based indexing technique that combines concepts with word stems. Using word stems can compensate for the incompleteness of the knowledge sources. Experimental results reveal that using phrase-based for indexing produces more accurate document retrieval than using word stems or concepts alone.

We also present an implementation that integrates the query expansion and phrase-based indexing in a medical digital library for the retrieval of patient records, laboratory reports and medical literature.

1. Introduction

Efficient document retrieval based on user query is achieved by indexing. Current technique uses word stem to index a document [1]. Such technique suffers from the inability to match words in query with their related words such as synonyms, hypernyms and hyponyms [2] in the documents. Therefore, there are recent attempts to index the document based on conceptual terms. However the knowledge sources are usually incomplete. As a result, past research reveals that although using conceptual terms for document indexing can solve some of the problems, it cannot outperform the word-stem-based model [3,4,5,6]. To remedy the incompleteness of the knowledge sources, we propose a phrase-based indexing model where we parse a document into phrases based on the conceptual terms in domain specific knowledge sources, and calculate the similarity between two documents using both the similarity between the concepts and the common word stems in them. Including word stem similarity in document similarity evaluation compensates for the incompleteness of the knowledge sources.

When seeking specific information regarding a particular topic, the user often has to pose a general query with concept terms. For example, not knowing that “contact lenses” is a treatment option for keratoconus, the user has to request for “treatment options for keratoconus” in the query. This results in low retrieval precision since documents are indexed by the specific terms. To remedy such shortcoming, we propose to substitute the general concept terms in the query with the specific terms. The level of relevancy of a specific term in the resulting query is determined by its co-occurrence with the general concept term, which can be mined from the corpus. Based on the query, the knowledge sources can identify the irrelevant conceptual terms and prevent them from being inserted into the query. Since the expanded queries match better with relevant documents, the document retrieval performance is improved.

We shall first present the phrase based index technique and the experimental results to show the performance improvements of phrased based index over word stem and concept based index methods, Next we shall present the knowledge based query extension technique and present the performance improvement derived from the query extension. Finally, we present an implementation of integrating the two proposed techniques in a medical digital library for retrieving medical textual records and reports.

2. Phrase-based indexing

To facilitate discussion, we shall use the following sample query in this section: “22 year old with hyperthermia, leukocytosis, increased intracranial pressure, and central herniation. Cerebral edema secondary to infection, diagnosis and treatment.” The first part of the query is a brief description of the patient; and the second part is the information need.

2.1 Word stem Indexing

A document is commonly represented as a vector of terms in a *vector space model* (VSM) [1]. The basis of the vector space corresponds to distinct terms in a document collection. Components of the document vector are the weights of the corresponding terms that represent their relative importance in the document. In a naïvest approach, we could treat a word as a term. Yet, morphological variants like "edema" and "edemas" are so closely related that they are usually conflated into a single *word stem*, e.g., "edem," by stemming [1,7]. Our sample query thus consists of word stems "hypertherm," "leukocytos," "increas," "intracran," "pressur," etc. Word stems are usually treated as notational, rather than conceptual entities. Two word stems are considered unrelated if they are different. For example, the stem of "hyperthermia" and that of "fever" are usually considered unrelated despite their apparent relationship. In stem-based VSM, word stems constitute the basis of the vector space. The base vectors are orthogonal to each other because different word stems are considered unrelated. The weight $w_{\alpha,u}^{s}$ of a word stem u in a document α is determined by the number of times u appears in α (known as the *term frequency*) and the number of documents that contain u (known as the *document frequency*) following the TF-IDF (term frequency, inverse document frequency) scheme [1]. In essence, the more often u appears in α, the more important u is in α. On the other hand, the more documents u belongs to, the less disambiguating power it has, and thus the less important it is.

Word stems are widely used as index terms. To improve retrieval accuracy, it is natural to replace word stems with concepts [3,4,5,6,8] or multiple-word combinations [9,10]. However, previous research showed not only no improvements, but degradation in retrieval accuracy when concepts were used in document retrieval [3,4,5,6] except when documents were very short [8]. When properly used, multiple-word combinations were shown to improve retrieval effectiveness for some special queries [9,10]. However, the retrieval effectiveness improvement for ad hoc queries is still questionable.

2.2 Concept-based VSM

Using word stems to represent documents results in the inappropriate fragmentation of concepts such as "increased intracranial pressure" into its component stems "increas," "intracran," and "pressur." Clearly, using *concepts* instead of single words or word stems as the vector space basis should produce a VSM that better mimics the human thought processes, and therefore should result in more accurate retrieval.

However, using concepts is more complex than using word stems. First, concepts are usually represented by multi-word phrases such as "increased intracranial pressure." More importantly, there exist synonymous and polysemous phrases. Two phrases sharing a concept are *synonymous*, and phrases that could represent more than one concept are *polysemous* [2]. For example, "hyperthermia" and "fever" are synonymous because they share the same concept "an abnormal elevation of the body temperature." At the same time, "hyperthermia" is polysemous, because in addition to the above meaning it also means "a treatment in which body tissues is exposed to high temperature to damage and kill cancer cells." Synonyms can be identified with the help of a dictionary or a thesaurus. Determining which concept a particular polysemous phrase represents is known as *word sense disambiguation* (WSD) [11]. Third, some concepts are related to one another. Hypernym and hyponym relations are important conceptual relations. If we say "an x is a (kind of) y" then concept x is a hyponym of concept y, and y is a hypernym of x [2]. "Hyperthermia" is a hyponym of "high body temperature;" and "high body temperature" is a hypernym of "hyperthermia."

Concept identifiers are usually used to identify concepts. Using UMLS [13] as a knowledge source, our sample query becomes (15967, 203597), (23518), and (151740) etc., representing "hyperthermia," "leukocytosis," and "increased intracranial pressure," etc., respectively.

In concept-based VSM, the basis of the vectors space consists of distinct concepts. To model the relationship of such concepts as "hyperthermia" and "elevated body temperature" we remove the orthogonality constraint on base vectors. Base vectors for two related concepts form an acute angle. Only when we cannot find any reasonable relations between two concepts that we treat their corresponding vectors as orthogonal. The cosine of the angle between two concept vectors is defined as the *conceptual similarity* between the corresponding concepts. The conceptual similarity thus ranges from 0 to 1 with 0 indicating unrelated and 1 indicating highly related concepts.

To study the effects of conceptual similarities, we shall compare two cases. In one case, we assume all different concepts are unrelated. Therefore, all base vectors of the vector spaces are orthogonal to one another. We label this case as "O" for orthogonal. In the other case, we derive conceptual similarities from knowledge sources. The resulting base vectors are no longer mutually orthogonal. We label this case as "NO" for non-orthogonal.

We derive the weight w_{α,x_i}^{c} of the i^{th} concept x_i in a document α using a slightly modified version of TF-IDF scheme. Higher weights are assigned to longer phrases that correspond to more specific concepts. For example, if the term frequencies and document frequencies for "increased intracranial pressure" and "hyperthermia" were identical, the former concept would obtain a higher weight than the latter.

2.2 Phrase-based VSM

Conceptual similarities needed in concept-based VSM are derived from knowledge sources. The quality of such VSM therefore depends heavily on the quality of the knowledge sources. The missing of certain conceptual relations in the knowledge sources could potentially degrades retrieval accuracy. For example, treating "cerebral edema" and "cerebral lesion" as unrelated is potentially harmful. Noticing their common of component word "cerebral" in the above phrases, we propose phrase-based VSM to remedy the incompleteness of the knowledge sources.

In phrase-based VSM, a document is represented as a set of phrases. Each phrase may correspond to multiple concepts (due to polysemy) and consist of several word stems. Our sample query now becomes [(15967, 203597), ("hypertherm")], [(23518), ("leukocytos")] and [(151740), ("increas", "intracran", "pressur")] etc.

Following the TF-IDF schemes in stem-based and concept-based VSMs, we can derive the stem weight $w^{s}_{\alpha,u_{i,k}}$ of the k^{th} stem $u_{i,k}$ and the concept weight $w^{c}_{\alpha,x_{i,m}}$ of the m^{th} concept $x_{i,m}$ in phrase i of α.

Similar to concept-based VSM, we study two cases, O and NO. In case O, different concepts are unrelated; while in case NO, concepts may be related. In both cases, distinct word stems are assumed to be unrelated.

2.3 Document Similarity

The similarity of two documents α and β is the cosine of the angle between their corresponding document vectors $\vec{\alpha}$ and $\vec{\beta}$; that is,

$$sim(\alpha,\beta) = \cos(\vec{\alpha},\vec{\beta}) = \frac{\vec{\alpha}\bullet\vec{\beta}}{\sqrt{\vec{\alpha}\bullet\vec{\alpha}}\sqrt{\vec{\beta}\bullet\vec{\beta}}} \tag{1}$$

We shall extend the vector dot product $\vec{\alpha}\bullet\vec{\beta}$ and denote the *extended dot product* (EDP) as $\vec{\alpha}\circ\vec{\beta}$ to represent the cases when the components of the vectors $\vec{\alpha}$ and $\vec{\beta}$ are related. Using the EDP in place of the dot product, we derive document similarity as,

$$sim(\alpha,\beta) = \frac{\vec{\alpha}\circ\vec{\beta}}{\sqrt{\vec{\alpha}\circ\vec{\alpha}}\sqrt{\vec{\beta}\circ\vec{\beta}}} \tag{2}$$

EDP Derivation

To derive the EDP in the phrase-based VSM, we first consider concepts without polysemy. Thus,

$$\vec{\alpha}\circ\vec{\beta} = \sum_{i,j} S^{c}_{i,j} \tag{3}$$

where $S^{c}_{i,j}$ is the conceptual contribution of phrase i in α and phrase j in β to the EDP. Assuming that each phrase represents a single concept, we have,

$$S^{c}_{i,j} = w^{c}_{\alpha,x_i} w^{c}_{\beta,y_j} s(x_i, y_j) \tag{4}$$

where $s(x_i, y_j)$ is the conceptual similarity between the i^{th} concept x_i in α and the j^{th} concept y_j in β. In the orthogonal case, $s(x,y)$ is reduced to the Kronecker delta function,

$$\delta(x,y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

In the non-orthogonal case, conceptual similarities are derived from knowledge sources.

Conceptual Similarity, S(x,y)

Given a hypernym hierarchy, the conceptual similarity $s(x, y)$ between two concepts x and y depends on both the distance between them in the hierarchy and their generality. When two concepts are farther apart in the hypernym hierarchy, they are less similar – a concept is less similarity to its grandparent than to its parent in the hypernym hierarchy. Thus we define the conceptual similarity to be inversely proportional to the number of "hops" between x and y, $d(x,y)$. The generality of a concept x can be derived from the number of all its descendants $D(x)$. The more descendants x has, the more general it is. A general concept like "disease" has much more descendants than a more specific concept like "hyperthermia" has. Because of the exponential growth of the number of descendants when a concept moves up a tree structure, we take the logarithm of the number of descendant in conceptual similarity calculation. The conceptual similarity is therefore defined to be inversely proportional to the logarithm of the number of descendants of the two. A final consideration is the boundary case when we reach the leaves of the hypernym tree. Let us assume we have two concepts x_o and y_o, where x_o is the only direct hypernym of y_o, y_o is the only hyponym of x_o, and y_o has no hyponym of its own. Concepts x_o and y_o are so much alike that we define the conceptual similarity between them to be c close to 1, say 0.9, to represent such closeness. As a result, the conceptual similarity between concepts x and y is,

$$s(x,y) = \frac{c}{d(x,y)\log_2(1 + D(x) + D(y))} \tag{5}$$

In order to use (4) in the presence of polysemy, we need to disambiguate senses. To avoid WSD cost, we use the most popular concept that a phrase represents as the meaning of the phrase. Alternatively, we derive the conceptual contribution to the similarity between two phrases using an aggregation of (4) over all possible concept pairs, where each pair consists of one concept from each phrase.

The contribution of word stems to the EDP is the sum of the weight product for those word stems common to both phrases,

$$S^{s}_{i,j} = \sum_{k,l} w^{s}_{\delta,u_{i,k}} w^{s}_{\theta,v_{j,l}} \delta\left(u_{i,k}, v_{j,l}\right) \qquad \textbf{(6)}$$

where $u_{i,k}$ and $v_{j,l}$ are the k^{th} word stem in phrase i in α and l^{th} word stem in phrase j in β respectively.

Given the contribution of concepts and stems, (4) and (6), we select the larger of the two as the contribution of phrase i in α and phrase j in β to the EDP and get,

$$\vec{\alpha} \circ \vec{\beta} = \sum_{i,j} \max\left(S^{c}_{i,j}, S^{s}_{i,j}\right) \qquad \textbf{(7)}$$

Such selection remedies the incompleteness of the knowledge sources. $\vec{\alpha} \circ \vec{\alpha}$ and $\vec{\beta} \circ \vec{\beta}$ can be derived similar to (7). The document similarity can then be computed from (2) using these EDPs.

2.4. Experimental Results

The Test Collection, OHSUMED

OHSUMED [12] is a large test collection used in many information retrieval system evaluations. The test set consists of a reference collection, a query collection, and a set of relevance judgments.

The reference collection is a subset of the MEDLINE database. Each reference contains a title, an optional abstract, a set of MeSH headings, author information, publication type, source, a MEDLINE identifier, and a sequence identifier. The query collection consists of 106 queries. Each query contains a patient description, an information request, and a sequence identifier. The sample query we use in this paper is query 57 in the collection. 14,430 references out of the 348K are judged by human experts to be not relevant, possibly relevant, or definitely relevant to each query.We use the title, the abstract, and the MeSH headings to represent each document; and the patient description, and the information request to represent each query.

The Knowledge Source

UMLS [13] is a medical lexical knowledge source and a set of associated lexical programs. The knowledge source consists of UMLS Metathesaurus, SPECIALIST lexicon, and UMLS semantic network. Especially of interest to us is its central vocabulary component – the Metathesaurus. It contains biomedical phrases from more than 60 vocabularies and classifications. The Metathesaurus contains 1.6M phrases representing over 800K concepts.

A concept unique identifier (CUI) identifies each concept. UMLS tends to assign a smaller CUI to a more popular sense of a phrase. Therefore, we use the concept with the smallest CUI in conceptual contribution calculation (2). Our experiment results show that such heuristic produces retrieval accuracy comparable to that produced by the aggregation approach where we consider all conceptual similarities due to different sense combinations from the phrases.

The Metathesaurus encodes many conceptual relations. We concentrate on hypernym relations. Two relations in UMLS roughly correspond to the hypernym relations: the RB (border than) and the PAR (parent) relations. For example, "hyperthermia" has a parent concept "body temperature change." We combine the 838K RB and 607K PAR relations into a single hypernym hierarchy.

Hypernymy is transitive [14]. For example, "sign and symptom" is a hypernym of "body temperature change" and "body temperature change" is a hypernym of "hyperthermia," so "sign and symptom" is also a hypernym of "hyperthermia." However UMLS Metathesaurus encodes only the direct hypernym relations but not the transitive closure. We derive the transitive closure of the hypernym relation and use (5) to calculate the conceptual similarities.

Phrase Detection

Given a set of documents (106 queries and 14K judged documents of OSHUMED), we need to detect any occurrences in a set of phrases (1.3M phrases in UMLS). We adopt the Aho-Corasick algorithm [15] for the set-matching problem to detect phrases:

First, Aho-Corasick algorithm detects *all* occurrences of any phrase in a document. But we only keep the longest, most specific phrase. For example, although both "edema" and "cerebral edema" are detected in the sample query, we keep only the latter and ignore the former.

Second, to detect multi-word phrases, we match stems instead of words in a document with UMLS phrases. We use Lovins stemmer [7] to derive word stems. To avoid conflating different abbreviations into a single stem, we define the stem for a word shorter than four characters to be the original word.

Third, stop-word removal is performed *after* the multi-word phrase detection. In this way, we correctly detect "secondary to" and "carcinoma" from "cerebral edema secondary to carcinoma." We would incorrectly detect "secondary carcinoma" if the stop-words ("to" in this case) were removed before the phrase detection.

Discussion of Results

To calculate retrieval accuracy using precision-recall [1], we combined the "possibly relevant" and "definitely relevant" judgments in OHSUMED into a single relevant category. Based on the type of VSM, we calculate the document similarity between each of the 14K documents and each of the 105 queries (one query does not have relevant document). For a given VSM and a query, we rank the documents from the most to the least similar to the query. When a certain number of documents are retrieved, *precision* is the percentage of retrieved documents that are relevant; and *recall* is the percentage of the relevant documents that has been retrieved so far. We evaluate the retrieval accuracy by interpolating the precision values at eleven recall points. The overall effectiveness of different VSM is then compared by averaging over the performance of all the 105 queries (Figure 1). The average of the eleven precision values gives an overview of the effectiveness of each VSM.

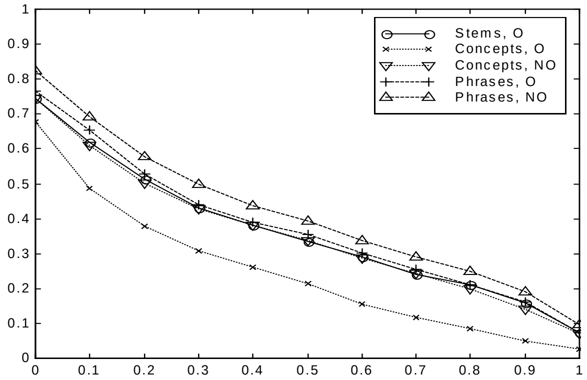

Figure 1. Comparison of the average precision-recall over 105 queries.

1. The baseline labeled (Stems, O) uses stem-based VSM. Its 11-point average precision is 0.363.
2. Considering the contribution of concepts only, and treating different concepts as unrelated (Concepts, O), we arrive at an 11-point average precision of 0.260, which is a 28% decrease from the baseline.
3. Similar to 2, but taking the concept inter-relationship into consideration (Concepts, NO), we achieve a significant improvement over 2. The average accuracy is similar to that of the baseline.
4. Considering contribution of both concepts and word stems in a phrase, but treating different concepts as unrelated (Phrases, O), we arrive at an 11-point average precision of 0.375, a 3% improvement over the baseline.
5. Similar to 4, but taking concept interrelations into consideration (Phrases, NO), we achieve an 11-point average precision of 0.420, which is a significant 16% improvement over the baseline.

Our experiment results reveal that viewing documents as concepts only and treating different concepts as unrelated can cause the retrieval accuracy to deteriorate (case 2). Considering concept inter-relations (case 3) or relating different phrases by their shared word stems (case 4) can both improve retrieval accuracy. The extended dot product combines contributions from the concepts and word stems. The phrase-based VSM utilizes such extended dot product and yields significant improvement in retrieval accuracy.

3. Enhance retrieval performance via query expansion

When posing a query, a user usually has a main objective (*key concept*, c_{key}) in mind and uses additional *general supporting conceptual* terms, c_s, to specify certain aspects of c_{key}. For example, when a user asks “Keratoconus, treatment options.”, “Keratoconus” (an eye disease) is the key concept whereas “treatment options” is a general supporting concept. Although such query is easy to form, it does not match well with relevant documents that use such specific supporting concepts as “contact lens” or “keratoplasty”.

To remedy this problem, we propose to substitute the general supporting concepts by specific concepts that used in the relevant documents. We need to select the set of specific concepts, and determine the weight of each of these concepts. We shall first present the weight determination method, and then compare two concept selection approaches.

For a specific concept *c*, its weight should represent the degree of correlation between *c* and the key concept term, c_{key}. For example, “contact lens” is a treatment option for “Keratoconus” but not “Back pain”, and therefore it should assign a larger weight in the expansion of “Keratoconus, treatment options” than that of “Back pain, treatment options”. We shall now present a scalable method for such weight assignment. We shall first represent concepts into inverted document vectors, and then use the similarity between the two inverted document vectors to represent the correlation between the two concepts.

Given a corpus of n documents, the inverted document vector for concept c_i, θ_{ci}, is defined as an n dimensional vector. The weight of component c_i in the vector represents the term frequency of concept c_i in each document. For example, if a corpus contains documents D_1, D_2 and D_3, and concept *c* occurs three, zero, and two times in these documents respectively, then θ_{ci} =<3, 0, 2>.

We further define the correlation between concepts c_i and c_j as:

$$correl\left(c_i,c_j\right)=\cos\left(\theta_{c_i},\theta_{c_j}\right)=\frac{\theta_{c_i}\bullet\theta_{c_j}}{\sqrt{\theta_{c_i}\bullet\theta_{c_i}}\sqrt{\theta_{c_j}\bullet\theta_{c_j}}}$$

The correlation between two concepts ranged from 0 to 1. For example, if θ_{c_i} =<3, 0, 2>, θ_{c_j} =<6, 0, 4>, and θ_{c_k} =<0, 1, 0>, then *correl*(c_i, c_j)=1 and *correl*(c_i,

c_k)=0. The correlation between all pairs of concepts can be computed offline and stored in a concept correlation table (see Figure 3). For query expansion, the weight assigned to c_i is the correlation between a supporting concept c_i and c_{key}.

There are two concept selection approaches: with or without knowledge sources. When no knowledge source is available, all the supporting concept terms that have zero correlation with the key concept, c_{key}, can be viewed as irrelevant concepts and filtered out. Let c_1, c_2, ..., c_k be all the concepts that have nonzero correlation with c_{key}. The expanded query becomes
$<(c_{key},1), (c_1, correl(c_1, c_{key})), \ldots, (c_k, correl(c_k, c_{key}))>$

In this expanded query vector, the higher the correlation value $correl(c_i, c_{key})$ is, the more emphasis is placed on the vector component of c_i. Clearly, the weight assigned to c_{key} is 1 since $correl(c_{key}, c_{key}) = 1$.

Since a general supporting concept in a query usually only relevant to one or two aspects of c_{key}, this motivate the use of knowledge source together with the key concept and the general supporting concept in the query to select the relevant set of specific supporting concepts. For example, "Keratoconus, treatment options" is emphasizing on the treatment options for the disease, instead of diagnosis methods or causes for the disease. ULMS [13] indicates only three categories of medical concepts as potential treatments: "Therapeutic and Preventive Procedures", "Medical Devices" and "Pharmalogical Substance". Thus only concepts that belong to these three categories will be expanded into the query. By filtering out the irrelevant supporting concepts, the computation complexity is greatly reduced. In addition, the precision in the low recall region increases. To evaluate the performance improvement of query expansion, we select 28 OHSUMED queries [12] that contain general supporting concepts as the test set. Our experimental results for the query set reveal the average query expansion size without knowledge source is 1227 terms per query, while using ULMS, the average expansion size reduced to 82.3 terms per query. This represents more than an order of magnitude reduction in query expansion size.

The retrieval performance improvement for the set of expaned OHSUMED queries is shown in Figure 2. We note that the expansion queries performed better than the non-expansion cases (base line). The query derived from knowledge-based expansion performs better than the cases without knowledge base in the low recall region, and the performance is reverse in the high recall region. This is because the non-knowledge based expansion case includes many irreverent specific concepts, which resulted in low precision in the low recall region. Since relevant documents in the high recall region are not typically covered by the highly correlated supporting concepts in the query, adding more terms by the non-knowledge based method yields better precision in that region. Aside from the computation complexity saving of using the knowledge based expansion method, the choice of which expansion method to use depends on the application. When low recall region is more of concern, then knowledge-based method is preferable; and if high recall region is more important, then the non-knowledge based method should be considered.

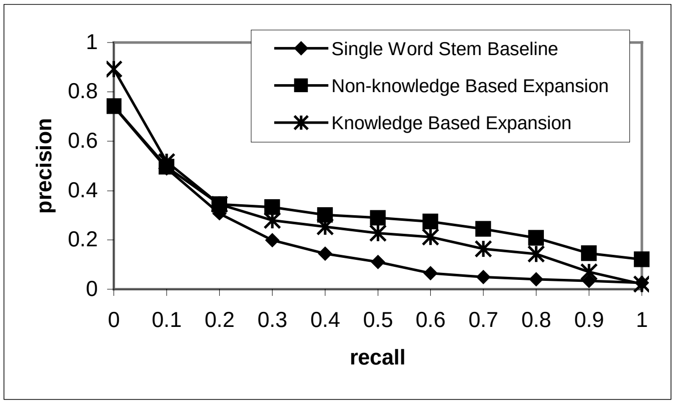

Figure 2. Retrieval performance improvements with query expansion

4. Applications

We shall now present a document retrieval system that integrates the knowledge-based query expansion and phrase-based indexing for a digital medical library. As shown in Figure 3, the system consists of three subsystems: a document indexing engine, a query expansion engine, and a document retrieval engine. The document indexing engine processes the knowledge source and the corpus offline and generates data structure necessary for the online query expansion and document retrieval. When the system receives a user query, the query expansion engine expands the query and the document retrieval engine then returns a set of documents relevant to the user.

The medical knowledge source, UMLS, is used in the system for phrase detection, conceptual similarity derivation, and expansion term filtering. The document indexing engine processes UMLS and the documents corpus separately. The conceptual similarity calculator derives the conceptual similarities between concepts from UMLS and stores them in the conceptual similarity table. The phrase detector identifies concepts in UMLS from the documents. The inverse document frequency calculator uses the output of the phrase detector to construct an inverse document frequency table. The phrase weight calculator in turn calculates the weights of concepts and word stems in each phrase, and converts the original corpus into a phrase-indexed corpus. The correlation calculator then derives the conceptual

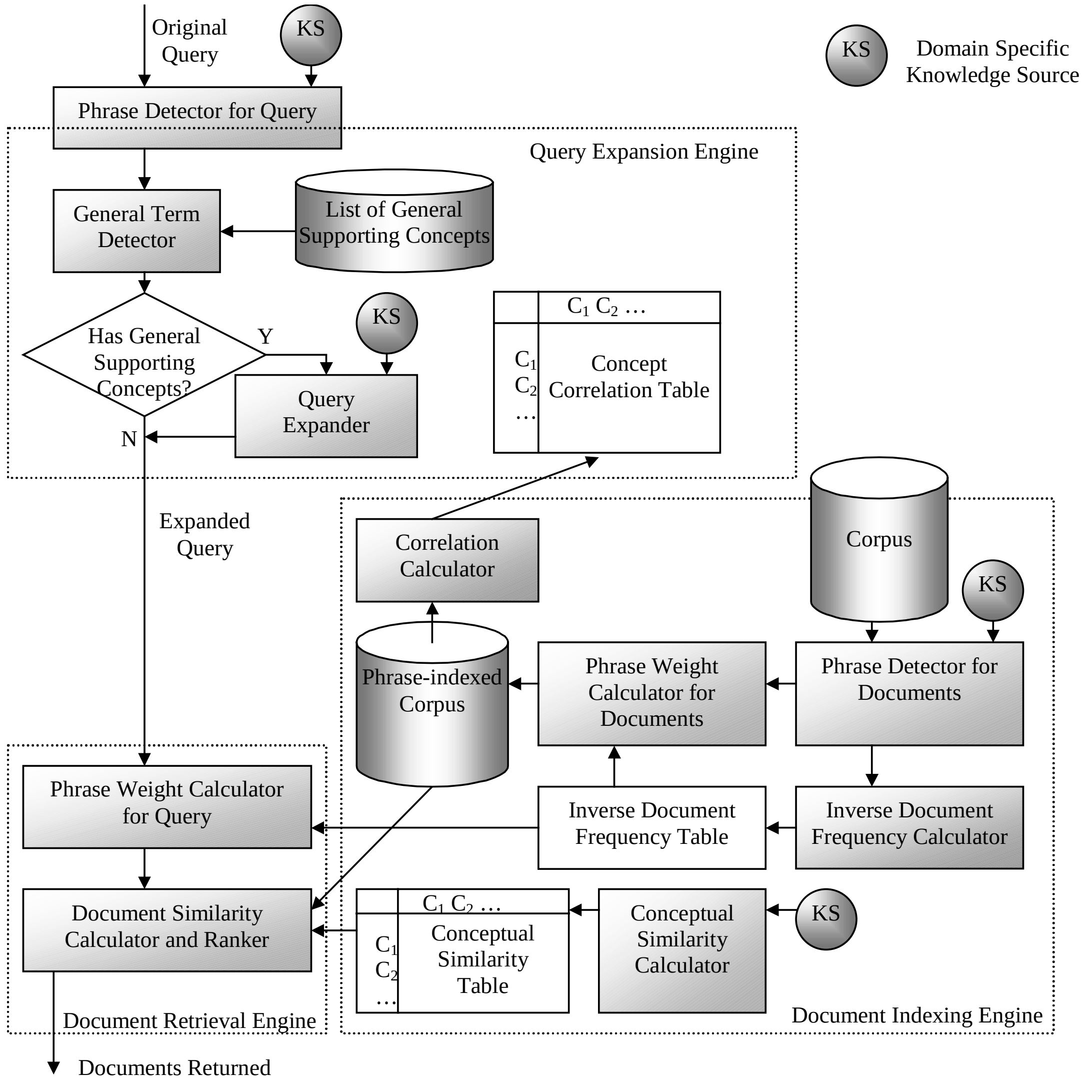
Original Query
KS
KS
Domain Specific Knowledge Source
Phrase Detector for Query
Query Expansion Engine
General Term Detector
List of General Supporting Concepts
Has General Supporting Concepts?
Y
N
KS
Query Expander
C_1 C_2 …
C_1
C_2
…
Concept Correlation Table
Expanded Query
Correlation Calculator
Corpus
KS
Phrase-indexed Corpus
Phrase Weight Calculator for Documents
Phrase Detector for Documents
Phrase Weight Calculator for Query
Inverse Document Frequency Table
Inverse Document Frequency Calculator
Document Similarity Calculator and Ranker
C_1 C_2 …
C_1
C_2
…
Conceptual Similarity Table
Conceptual Similarity Calculator
KS
Document Retrieval Engine
Document Indexing Engine
Documents Returned

Figure 3. A phrase based indexing and query expansion document retrieval system.

correlations from the phrase-indexed corpus and stores them in the concept correlation table.

When the system receives a user query, phrase detector first parse the query into phrases. The general term detector then determines if any query expansion is necessary by consulting the general supporting concept list. If no general supporting concepts are detected. The original query is directly input into the document retrieval engine. Otherwise, the query expander replaces general concept terms with the specific ones in the appropriate categories as specified in UMLS. The weights of the expanded concepts are looked up from the concept correlation table.

The document retrieval engine compares the expanded query with the phrase-index and returns a set of documents to the user. First, the similarity calculator consults the conceptual similarity table to calculate the phrase-based document similarities. The ranker then returns to the user those documents the most similar to the query.

5. Conclusion

We introduced a knowledge-based technique to rewrite a user query containing general conceptual terms to one containing specific terms that related to the general supporting concept. These specific supporting terms are mined from the corpus. Experimental results show that retrieval using such expanded queries is more effective than the original queries. Because the additional concept terms included in the expanded query are selected from few categories as indicated by the knowledge source, the average size of the expanded queries in our approach is much smaller (less than an order of magnitude) than that produced by the full query expansion technique and also yield better retrieval performance in the low recall region which is of interest to most applications.

We developed a new vector space model that uses phrases to represent documents. Each phrase consists of multiple concepts and words. Similarity between two phrases is jointly determined by the conceptual similarity and their common word stems. Our experimental result reveals that the phrase-based VSM yields a 16% increase of retrieval effectiveness than the stem-based VSM. This improvement is because multi-word concepts are natural unit of information and using word stems in phrase-based document similarity compensates for the inaccuracy in conceptual similarities derived from incomplete knowledge sources.

We also presented an implementation that integrates the above techniques into a digital medical library at UCLA for the retrieval of patient records, laboratory reports and medical literatures.

References

[1] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*, 1983

[2] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller. Introduction to WordNet: an On-line Lexical Database. In *WordNet: an Electronic Lexical Database*, 1-19, 1998

[3] M. Mitra, C. Buckley, A. Singhal and C. Cardie. An Analysis of Statistical and Syntactic Phrases. In *Proc. RIAO97*, 200-214, 1997

[4] R. Richardson and A.F. Smeaton. Using WordNet in a Knowledge-based Approach to Information Retrieval. In *Proc. 17th BCS-IRSG*, 1995

[5] M. Sussna. Text Retrieval using Inference in Semantic Matanetworks. *PhD Thesis*, University of California, San Diego, 1997

[6] E.M. Voorhees. Using WordNet to Disambiguate Word Sense for Text Retrieval. In *Proc. 16th ACM-SIGIR.*, 171-180, 1993

[7] J.B. Lovins. Development of a Stemming Algorithm. In *Mechanical Translation and Computational Linguistics*, 11(1-2), 11-31, 1968

[8] A.F. Smeaton and I. Quigley. Experiments on using Semantic Distances Between Words in Image Caption Retrieval. In *19th Proc. ACM-SIGIR*, 174-180, 1996

[9] D. Johnson, W.W. Chu, J.D. Dionisio, R.K. Taira and H. Kangarloo. Creating and Indexing Teaching Files from Free-text Patient Reports. In *AMIA'99*, 1999

[10] J.A. Goldman, W.W. Chu, D.S. Parker and R.M. Goldman. Term Domain Distribution Analysis: A Data Mining Tool for Text Databases. In *2001 IMIA Yearbook of Medical Informatics*, 96-101, 2001

[11] N. Ide and J. Véronis. Word Sense Disambiguation: the State of the Art. In *Computational Linguistics*, 24(1), 1-40, 1998

[12] W. Hersh, C. Buckley, T.J. Leone and D. Hickam. OHSUMED: an Interactive Retrieval Evaluation and New Large Test Collection for Research. In *Proc. 22nd ACM-SIGIR Conf.*, 191-197, 1994

[13] National Library of Medicine. *UMLS Knowledge Sources, 12th edition*, 2001

[14] J. Lyons. *Semantics*, 1977

[15] A.V. Aho and M.J. Corasick. Efficient String Matching: an Aid to Bibliographic Search. In *CACM*, 18(6), 330-340, 1975