Information Retrieval and Clustering
W. Wu, H. Xiong and S. Shekhar(Eds.) pp. – - –

Techniques for Textual Document Indexing and Retrieval via Knowledge Sources and Data Mining

Wesley W. Chu
Computer Science Department
University of California, Los Angeles CA 90024
E-mail: wwc@cs.ucla.edu

Victor Zhenyu Liu
Computer Science Department
University of California, Los Angeles CA 90024
E-mail: vicliu@cs.ucla.edu

Wenlei Mao
Computer Science Department
University of California, Los Angeles CA 90024
E-mail: wenlei@cs.ucla.edu

Contents

Abstract

We present index techniques for retrieval of textual documents. We shall first discuss techniques for indexing document without domain knowledge, and then present techniques using domain knowledge. A new phrase-based indexing technique that combines concepts with word stems is introduced. Using word stems can compensate for the incompleteness of the knowledge sources. Experimental results reveal that using phrase-based indexing produces more accurate document retrieval than using word stems or concepts alone.

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 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 that answers user query is achieved by indexing. The current technique uses word stem to index a document [1]. Such a technique suffers from the inability to match words in a 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 deficiency 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 stems in addition to concepts in document similarity evaluation compensates for the shortcoming of using concept terms alone caused by 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 a 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 inserting 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 indexing technique and the experimental results to show the performance improvements of phrased based indexing over word-stem- and concept-based indexing 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 Indexing without domain knowledge

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 naïve 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 $w^s_{\alpha,u}$ 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.

2.2 Multi-word indexing

Word stems are widely used as index terms. To improve retrieval accuracy, multi-word index terms may be used for indexing.

Several methods exist for defining multi-word indexing terms. An n-gram is defined as an ordered sequence of n words taken from a document. For example, "several methods" and "methods exist" are the first two bi-grams of the last sentence. Given a document d, by providing context lacking from isolated words, n-grams may more accurately model the content of documents.

Two other factors may influence the effectiveness of n-grams as indexing terms. First, n-grams depend on word order, thus "right upper lobe mass" is not equivalent to "mass right upper lobe." Second, n-grams are limited by word proximity, requiring that words appear next to one another in the original text. For example, in the text sample: "a mass is seen in the right upper lobe," here, "mass" and "right upper lobe" will only appear together if a 8-gram is used to model the text. Removing typical stop words from the sample results in “mass seen right upper lobe,” still for the finding and anatomy descriptions to appear together requires an n-gram with a minimum length of 5 terms.

N-grams [8] may improve retrieval precision by providing additional context over isolated words. However, reliance on the original document's word order as well as word proximity may decrease retrieval recall and may require longer n-grams to be used to model documents.

We define an n-word combination as an unordered collection of n words taken from a document [9]. Given the text sample: “right upper lobe mass,” there are 6 different 2-word combinations, including “right upper” and ”upper mass.” Unlike n-grams, n-word combinations (n-combos) do not depend on word order or proximity. Any set of n words can form an n-combo.

Removing the restriction on word order and proximity dramatically increases the number of potential n-combos. Given a document d of length l, there are $l!/(n!(l-n)!)$ n-word combinations in d. As the length of the documents grows, the number of n-combos grows dramatically (e.g., a 100 word document has the potential of 3,921,225 4-combos, a 200 word document has 64,684,950). Brute force calculation of all possible n-word combinations in a document, even for relatively small n, is too time and space expensive. In order to use n-word combinations, some method of limiting the search space must be defined. Furthermore, a method to select which n-combos should be used as indexing terms must be developed.

Although each document has a central theme (e.g., medical report describes an individual patient), the concepts useful for indexing are described in the individual sentences of the document. Limiting the search scope to individual sentences will dramatically decrease the time and space required to calculate n-word combinations, while focusing on relevant indexing terms. Furthermore, stop-word lists are employed to further reduce the search space by factoring out those words that do not carry any semantic significance.

New efficient algorithms based on pattern decomposition (PD) technique have been developed [10] to improve computation efficiency by an order of magnitude in finding the frequent item set as compared to the Aprior algorithm [11]. The PD algorithm makes the n-word combo indexing computation feasible.

When properly used, multiple-word combinations were shown to improve retrieval effectiveness for some special queries[9]. However, the retrieval effectiveness improvement for ad hoc queries is still questionable.

3 Indexing with domain knowledge

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." It is natural to replace word stems with concepts. Clearly, using concepts instead of single words or word stems as the vector space basis should produce a VSM that better mimics human thought processes, and therefore should result in more accurate retrieval. 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 [12].

This is because 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 tissue 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) [13]. 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]. For example, "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 [14] 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 vector 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. It is 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 synonymous 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. In the other case, we derive conceptual similarities from knowledge sources. The resulting base vectors are no longer mutually orthogonal.

We derive the weight w^c_{α,x_i} 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.

3.1 Phrase-based VSM[15]

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 absence of certain conceptual relations in the knowledge sources could potentially degrade retrieval accuracy. For example, treating "cerebral edema" and "cerebral lesion" as unrelated is potentially harmful. Noticing their common component word of "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 $w^s_{\alpha,x_{i,m}}$ of the m^{th} concept $x_{i,m}$ in phrase i of α.

Similar to concept-based VSM, we study two cases. In one, different concepts are treated as unrelated; in the other, concepts may be related. In both cases, distinct word stems are assumed to be unrelated.

3.2 Document similarity

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

$$\mathrm{sim}(\alpha, \beta) = \cos(\vec{\alpha}, \vec{\beta}) = \frac{\vec{\alpha} \cdot \vec{\beta}}{\sqrt{\vec{\alpha} \cdot \vec{\alpha}}\sqrt{\vec{\beta} \cdot \vec{\beta}}}$$

To calculate phrase-based document similarity, we shall extend the document vector dot product $\vec{\alpha} \cdot \vec{\beta}$ and denote the *extended dot product* (EDP) as $\vec{\alpha} \circ \vec{\beta}$. Using the EDP in place of the dot product, we derive document similarity as,

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

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

$$\vec{\alpha} \circ \vec{\beta} = \sum_{i,j} S^c_{i,j}$$

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,

$$S^c_{i,j} = w^c_{\alpha,x_i} w^c_{\beta,y_j} s(x_i, y_j) \tag{2}$$

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. When different concepts are treated as unrelated, $s(x, y)$ in (2) is reduced to the Kronecker delta function,

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

When concepts may be related, we derive conceptual similarities from knowledge sources.

In order to use (2) 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 phrase's meaning. Alternatively, we derive the conceptual contribution to the similarity between two phrases using an aggregation of (2) over all possible concept pairs. Each pair consists of one concept from each phrase.

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

$$S^s_{i,j} = \sum_{k,l} w^s_{\alpha,u_{i,k}} w^s_{\beta,v_{j,l}} \delta(u_{i,k}, v_{j,l}) \tag{3}$$

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 (2) and stems (3), we select the larger of the two as the contribution of phrase i in α and phrase j in β to the EDP. Furthermore, we assign different similarity contribution factors f^c and f^s to the concept and stem contributions respectively, to indicate the relative importance of each contribution. Thus,

$$\vec{\alpha} \circ \vec{\beta} = \sum_{i,j} \max(f^c S^c_{i,j}, f^s S^s_{i,j}) \tag{4}$$

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 (4). The document similarity can then be computed from (1) using these EDPs. Because we use an identical EDP formula to calculate the denominator and numerator in (1), the ratio of the similarity contribution factors f^c/f^s, instead of their absolute values, which is important. Changing the values of f^c and f^s while maintaining their ratio f^c/f^s produces identical document similarity $\text{sim}(a, b)$.

Conceptual Similarity in Hypernym Hierarchy Given a hypernym hierarchy, the conceptual similarity $s(x, y)$ between 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 similar 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 its descendants $D(x)$. The more descendants x has, the more general it is. A general concept like “disease” has many 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 descendants 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_0 and y_0, where x_0 is the only direct hypernym of y_0, y_0 is the only hyponym of x_0, and y_0 has no hyponym of its own. Concepts x_0 and y_0 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,

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

3.3 Experimental results

The Test Collection, OHSUMED OHSUMED [16] 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; the patient description and the information request represent each query.

The Knowledge Source UMLS [14] 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. Because of synonymy, multiple phrases can be associated with one CUI. For example, 71 phrases in 15 languages are associated with CUI 15967. Some English phrases for this CUI are "fever," "hyperthermia," "high body temperature," and "temperature, high." UMLS tends to assign a smaller CUI to the more popular sense of a phrase. For example, the CUI for the "high body temperature" sense of "hyperthermia" is 15967, while the CUI for its "treatment" sense is 203597. Therefore, we use the concept with the smallest CUI in conceptual contribution calculation (2). Our experimental 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 pairs of relations in UMLS roughly correspond to the hypernym relations: the RB/RN (broader than/narrower than) and the PAR/CHD (parent/child) relations. For example, "hyperthermia" has a parent concept "body temperature change." The PAR and CHD relations are redundant. If (x, y) is in PAR, then (y, x) is in CHD; and vice versa. Similarly, the RB and RN relations are redundant. Therefore, we combine the 838K RB and 607K PAR relations into a single hypernym hierarchy.

Hypernymy is transitive [17]. 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 OHSUMED), we need to detect any occurrences in a set of phrases (1.3M phrases in UMLS). We adopt the Aho-Corasick algorithm [18] 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.

1. The baseline (Stems) uses stem-based VSM. Its 11-point average precision is 0.375.

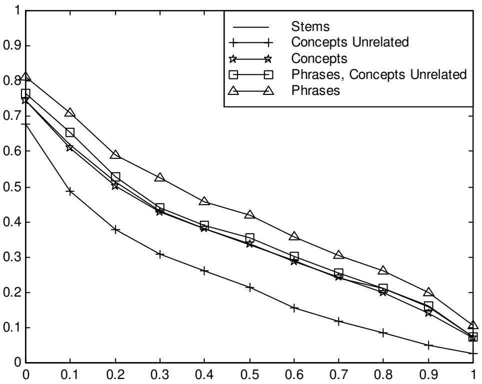

Figure 1: Comparison of the average precision-recall over 105 queries.

2. Considering the contribution of concepts only, and treating different concepts as unrelated (Concepts Unrelated), we arrive at an 11-point average precision of 0.270, which is a 28% decrease from the baseline.
3. Similar to 2, but taking the concept inter-relationship into consideration (Concepts), 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, Concepts Unrelated), we arrive at an 11-point average precision of 0.386, a 3% improvement over the baseline. In both this case and the following case 5, we use equal similarity contribution factors $f^s = f^c$.
5. Similar to 4, but taking concept interrelations into consideration (Phrases), we achieve an 11-point average precision of 0.431, which is a significant 15% 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 products and yields significant improvement in retrieval accuracy. To study the relative importance of the concept contribution (2) and stem contribution (3) to the EDP, we vary the similarity contribution factors f^c and f^s in (4) and calculate the average of the 11-point average precisions over the 105 queries. Varying from the stem-only case $(f^s, f^c) = (10, 0)$ to the concept-only

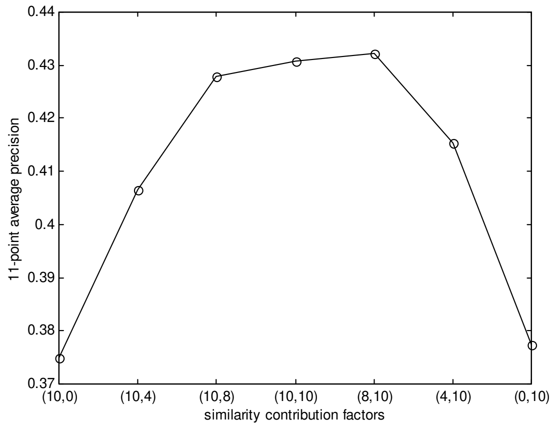

Figure 2: Effect of similarity contribution factors (f^s,f^c) on the average precision for phrase-based indexing.

case $(f^s, f^c) = (0, 10)$, we obtain the effect of the similarity contribution factors on the average precision for phrase-based VSM (Figure 2). Note that the average precision is relative constant in the range of $(f^s, f^c) = (10, 8)$ to $(f^s, f^c) = (8, 10)$. As a result, we use equal similarity contribution factor $f^s = f^c$ to obtain best average precision results. Thus (4) reduces to

$$\vec{\alpha} \circ \vec{\beta} = \sum_{i,j} \max(S^c_{i,j}, S^s_{i,j})$$

The 11-point average precision for the stem-only case $(f^s, f^c) = (10, 0)$ in Figure 2 is the average of the eleven precision values for case 1 (Stems) in Figure 1. Similarly, the concept-only case $(0, 10)$ in Figure 2 corresponds to case 3 (Concepts) in Figure 1; and the equal similarity contribution factor case $(10, 10)$ in Figure 2 corresponds to case 5 (Phrases) in Figure 1.

4 Improving retrieval performance via query expansion

In real application, a user's query often consists of two parts: a key concept, c_{key}, which is the user's main objective in mind, and general supporting concepts, c_s, which specify certain aspects of c_{key}. For example, in a query "Keratoconus, treatment options", "Keratoconus" (an eye disease) is the key concept whereas "treatment options" is a general supporting concept. This kind of query formation is typical in the 106 OHSUMED queries [16], some of which are shown in Table 1. Also a recent study [19, 20], reveals that doctors' clinical questions about patient care can be generalized into a limited number of categories. Seven out of the top ten categories (a 61.68% coverage over all questions studied) used this formation of a key concept plus general supporting concepts. Although such techniques are easy to form queries, the queries do not match well with relevant documents that use specific supporting concepts. In the example above, documents discussing treatment options for keratoconus use specific terms such as "contact lens" or "penetrating keratoplasty" instead of the general terms of "treatment" or "option". To remedy this problem, we propose to substitute the general supporting concepts with specific concepts used in the relevant documents. The rewritten query matches better with the relevant documents, thus improving the retrieval effectiveness. In this section, we will first review the past research on query expansion. Second, we will propose a knowledge-based approach that leads to more focused query expansion than traditional methods. The weight assigning scheme is discussed in 4.3. Finally we will present experimental results comparing the knowledge-based query expansion with basic expansion approaches.

4.1 Previous research on automatic query expansion

Using automatic query expansion to solve the query-document mismatch problem has been studied for decades [21, 22, 23, 24, 25, 26]. The basic idea is to first compute all terms that highly co-occur with the original query terms, and then append them to the original query. Co-occurrence measures are obtained automatically from the corpus from which documents are retrieved. It is assumed that terms highly co-occurring with the original query terms are likely to be synonyms or semantically related. Therefore, adding such extra terms is supposed to help retrieve relevant documents that do not use the exact terms in the original query. However, in early works by Sparck Jones [21, 22], the effectiveness of query expansion was unable to be justified. Not until the early nineties when large corpuses became common in research, and when the vector model and different vector weighting schemes had been intensively studied [1, 27], were researchers able to show consistently improved retrieval effectiveness via query expansion [23, 24, 25]. In [23], word stems are used to index queries and documents. A stem is represented by a vector of documents in which each component is weighted using the normalized $tf * idf$ measure [27]. Using VSM, the co-occurrence measure between two stems is computed. The weight for an expanded stem is the average co-occurrence measure between that

Table 1: Sample OHSUMED queries which each contain a key concept and general supporting concepts. Key concepts are shown in capital letters and general supporting concepts are in italics

Query ID	*Original Query Form*
13	**LACTASE DEFICIENCY** *therapy options*
16	**CHRONIC FATIGUE SYNDROME**, *management and treatment*
37	**FIBROMYALGIA / FIBROSITIS**, *diagnosis and treatment*
38	**DIABETIC GASTROPARESIS**, *treatment*
42	**KERATOCONUS**, *treatment options*
43	**BACK PAIN**, *information on diagnosis and treatment*
47	**URINARY RETENTION**, *differential diagnosis*
53	**LUPUS NEPHRITIS**, *diagnosis and management*

stem with each respective original query stem. Considering term co-occurrence on an entire document basis may be too coarse. To remedy this problem, [24] proposed an approach in which two terms are considered as co-occurring only when they appear close enough to each other (e.g. in one paragraph or in the distance of a few sentences). Using the same dataset as [24], this method obtained better retrieval accuracy. Combining this idea of co-occurrence computation with the blind feedback method further improved retrieval performance [25].

4.2 A knowledge-based approach to selectively expand concepts that are relevant to the original query

Most of the previous studies used a stem-based approach. In other words, queries are indexed and expanded by word stems. In these studies, stems are not associated with any domain knowledge. As a result, it is impossible to distinguish stems that are truly semantically related to the original query from those that are not, even though they all highly co-occur with the original query stems. Also, it is hard to comprehend stems that are expanded into the query, making it even harder to refine the expansion operation. For the sample query of "Keratoconus, treatment options", we computed 10 word stems that are expanded into the query with the highest weights (Table 2). The experimental settings were the same as [23]. As can be seen from the

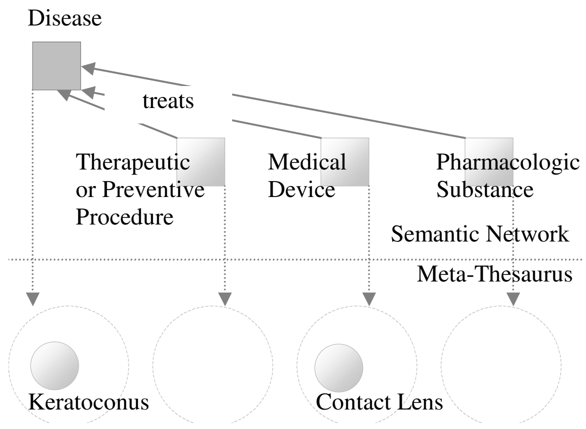

Figure 3: Mapping from concepts to semantic types and the relationships among semantic types in UMLS

table, not all components added to the query are related to “treatment options”, resulting in a less focused expansion. Therefore we propose a knowledge-based approach to obtain more focused query expansion. For our study, we used UMLS as the knowledge source and OHSUMED as the corpus. UMLS consists of three parts, the Metathesaurus, the Semantic Network and the SPECIALIST Lexicon. A group of concepts in the Metathesaurus are abstracted into one semantic type in the Semantic Network. Semantic types are associated with each other by a limited number of relationships. Figure 3 shows that “Keratoconus” and “Contact lens” belong to the semantic types of “Disease or Syndrome” and “Therapeutic or Preventive Procedure”, respectively; “Procedure” is one of the types that “treats” “Disease”. However, whether there is a “treats” link between “Keratoconus” and “Contact Lens” is not indicated by UMLS.

In the knowledge-based query expansion approach, we first assume that the key concept section of the query can be marked out by the user, and also the general supporting concepts are represented in terms of relationships among semantic types. Given the previous study on clinical question commonalities, and the improved completeness coverage of UMLS from version to version, this is a reasonable assumption. Second, the concept unique identifier (CUI) of the key concept can be automatically detected from the marked out section. Third, we abstract the CUI into its corresponding semantic type (e.g. “Disease or Syndrome” for “Keratoconus”), and follow the relationships as indicated by the general supporting concepts to a set of relevant semantic types. Finally, only concepts that belong to these relevant semantic types are considered as relevant specific supporting concepts and are

Table 2: Top 10 stems that are expanded into the sample query "Keratoconus, treatment options"

Ranking based on weights assigned	*Word stem*
1	corne
2	keratoplast
3	acu
4	visu
5	epikeratoplast
6	lens
7	keratometer
8	penetr
9	len
10	ey

added to the query. We also add the key concept's parents, children and siblings (all children of the parents of the key concept) to match relevant documents that use a broader or narrower topic. The scope is restricted to be one level up and down from the key concept, mainly because experimental results show that adding more ancestors or descendants hardly affects retrieval precision. Table 3 shows the top 10 concepts expanded into the sample query "Keratoconus, treatment options" using this knowledge-based approach. Compared to table 2, we note the knowledge-based query expansion is much more focused on the treatment aspect. In addition, the knowledge-based approach is able to filter out irrelevant specific supporting concepts. As a result, the expansion size is much smaller, which greatly reduced the computation complexity.

4.3 Weight assignment for expanded vector components

For a specific supporting 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". Let us 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 con-

Table 3: Top 10 concepts that are expanded into the sample query using the knowledge-based approach

Ranking based on weights assigned	*Concept c_i*
1	Cornea Transplantation
2	Contact lens
3	Penetrating keratoplasty
4	Epikeratoplasty
5	Epikeratophakia
6	Eyeglasses
7	Buttons
8	Radial Keratotomy
9	Trephines
10	Thermokeratoplasty

cepts. Given a corpus of n documents, the inverted document vector for concept c_i, $\vec{\theta_{c_i}}$, is defined as an n-dimensional vector. Each component in this vector corresponds to a document, the weight of which represents the term frequency of concept c_i in that particular 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 $\vec{\theta_{c_i}} =< 3, 0, 2 >$.

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

$$\text{correl}(c_i, c_j) = \cos(\vec{\theta_{c_i}}, \vec{\theta_{c_j}}) = \frac{\vec{\theta_{c_i}} \cdot \vec{\theta_{c_j}}}{\sqrt{\vec{\theta_{c_i}} \cdot \vec{\theta_{c_i}}}\sqrt{\vec{\theta_{c_j}} \cdot \vec{\theta_{c_j}}}} \tag{6}$$

The correlation between two concepts ranged from 0 to 1. For example, if $\vec{\theta_{c_i}} =< 3, 0, 2 >, \vec{\theta_{c_j}} =< 6, 0, 4 >$, and $\vec{\theta_{c_k}} =< 0, 1, 0 >$, 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. For query expansion, the weight assigned to c_i is the correlation between a supporting concept c_i and c_{key}.

4.4 Experimental results

To evaluate the performance improvement of knowledge-based query expansion, we select 28 OHSUMED queries [16] that contain general supporting concepts as treatment and diagnosis options (see Table 1 for examples). We followed the procedure in [23] to generate the result

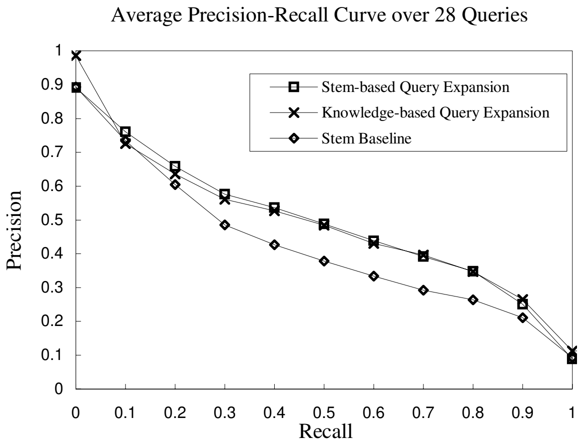

Figure 4: Retrieval performance improvements with knowledge-based query expansion

for stem-based query expansion. We used the same stem-based VSM as in 3.4 to generate the baseline.

Our experimental results for the query set reveal the average query expansion size using traditional stem-based expansion is 14584 terms per query, while using UMLS, the average expansion size reduced to 235 terms per query. This represents more than an order of magnitude reduction in query expansion size.

The retrieval performance comparison for the set of selected OHSUMED queries is shown in Figure 4. We note that both expansion approaches obtain higher precision than the baseline, and the knowledge-based approach performed better than the stem-based approach in the low recall region (below 10% recall).

Knowledge-based query expansion using concept-based VSM can significantly improve the computation complexity (more than order of magnitude) over the full stem-based expansion. However, due to the insufficient knowledge granularity, the retrieval performance is not uniformly better than the traditional stem-based approach, particularly in the high recall region. Therefore, we are currently evaluating the knowledge-based query expansion using stem-based VSM. Our preliminary results reveal that knowledge-based query expansion with stem-based VSM has uniformly better retrieval performance than that of the traditional stem-based query expansion, but with less significant complexity reduction (an order magnitude of reduction). Thus, the user is provided with a choice of query expansion methods with different levels of computation and performance tradeoffs. We are also

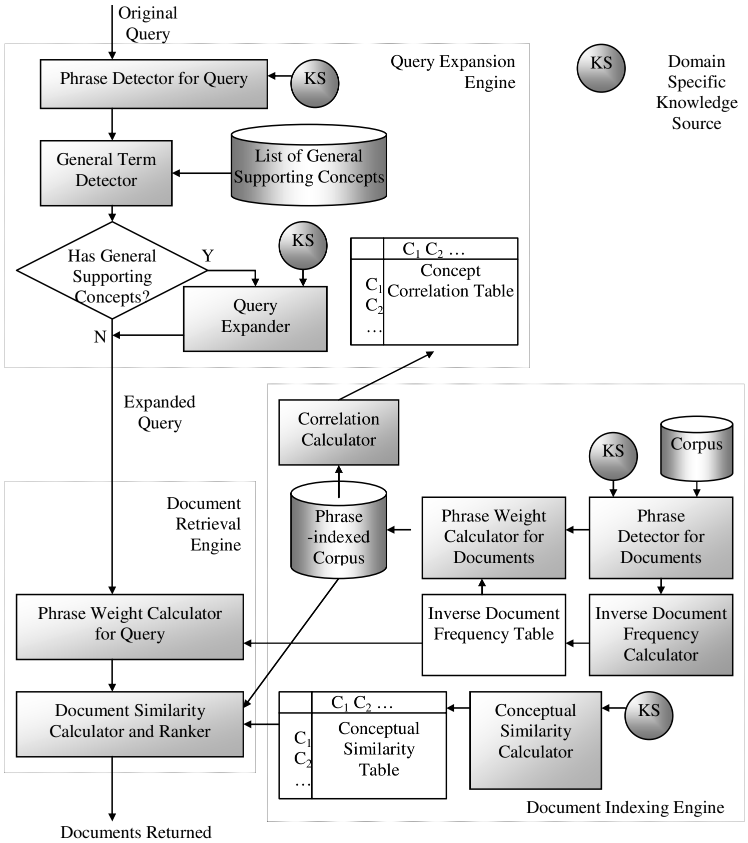

Figure 5: A phrase based indexing and query expansion document retrieval system

planning to investigate the knowledge-based query expansion using phrase-based VSM which will yield comparable performance as that of the knowledge-based expansion using stem-based VSM but with lower computation complexity.

5 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 5, 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 the 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 query. 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 document 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 correlations from the phrase-indexed corpus and stores them in the Concept Correlation Table. When the system receives a user query, Phrase Detector first parses 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 which are the most similar to the query.

6 Summary

We have presented indexing techniques for the retrieval of textual documents. First, we presented indexing techniques without domain knowledge, such as word stem and multi-word indexing and their shortcomings. Next we discussed indexing with domain knowledge of the corpus and 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 15% increase of retrieval effectiveness over that of the stem-based VSM. This improvement is because multi-word concepts are natural units of information, and using word stems in phrase-based document similarity compensates for the inaccuracy in conceptual similarities derived from incomplete knowledge sources.

We introduced a knowledge-based technique to rewrite a user query containing general conceptual terms into one containing specific terms. These specific supporting concepts are selected via knowledge sources and are related to the general supporting concept and the query's key concept. Weights for those specific concepts are assigned from datamining the corpus. Experimental results show that retrieval using such expanded queries is more effective than the original queries. The average size of the expanded queries in the knowledge-based approach is much smaller (reduced by more than orders of magnitude) than that produced by the stem-based query expansion, and also yield better retrieval performance in the low recall region which is of interest to most applications. 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 literature.

References

[1] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, (McGraw-Hill Inc., 1983).

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

[3] M. Mitra, C. Buckley, A. Singhal and C. Cardie, An Analysis of Statistical and Syntactic Phrases, *Proceedings of the Fifth RIAO Conference*, (1997), pp. 200-214.

[4] R. Richardson and A.F. Smeaton, Using WordNet in a Knowledge-based Approach to Information Retrieval, *Proceedings of the 17th BCS-IRSG Colloquium on Information Retrieval*, (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 Proceedings of the 16th Annual ACM SIGIR ConferenceonResearch and Development in Information Retrieval*, (1993), pp. 171-180.

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

[8] L.P. Jones, Jr. E.W. Gassie and S. Radhakrishnan, INDEX: The statistical basis for an automatic conceptual phrase-indexing system. *In Journal of American Society for Information Science*, 41(2) (1990), pp. 87-97.

[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] Q. Zou, W.W. Chu, D.B. Johnson and H. Chiu. Pattern decomposition algorithm for mining frequent patterns. *In Journal of Knowledge an Information System*, 4(4), (2002).

[11] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *In VLDB'94*, (1994), pp. 487-499.

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

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

[14] National Library of Medicine. UMLS Knowledge Sources, 12th edition, (2001).

[15] W. Mao and W.W. Chu. Free text medical document retrieval via phrased-based vector space model. *Proc. AMIA'02*, (2002).

[16] 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.*, (1994), pp. 191-197.

[17] J. Lyons. Semantics, (1977).

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

[19] J.W. Ely, J.A. Osheroff, M.H. Ebell, G.R. Bergus, et al. 1999. Analysis of questions asked by family doctors regarding patient care. *British Medical Journal*, 319:358-361, 1999

[20] J.W. Ely, J.A. Osheroff, P.N. Gorman, M.H. Ebell, et al. 2000. A taxonomy of generic clinical questions: classification study. *British Medical Journal*, 321:429-432, 2000

[21] K. Sparck Jones. Automatic keyword classification for information retrieval. Butterworth, London, 1971

[22] K. Sparck Jones. Collecting properties influencing automatic term classification. *Information Storage and Retrieval*, 9:499-513, 1973

[23] Y. Qiu and H.P. Frei. Concept-based query expansion. In *Proc. 16th ACM-SIGIR*, pages 160-169, 1993

[24] Y. Jing and W.B. Croft. An association thesaurus for information retrieval. In *Proc. RIAO'94*, pages 146-160, 1994

[25] J. Xu and W.B. Croft. Query expansion using local and global document analysis. In *Proc. 19th ACM-SIGIR*, pages 4-11, 1996

[26] E.N. Efthimiadis. Query expansion. In *Annual Review of Information Science and Technology*, 31:121-187, 1996

[27] G. Salton and C. Buckley, 1988. Term weighting approaches in automatic text retrieval, *Information Processing & Management*, 24(5):513-523, 1988