Keyphrase Extraction-Based Query Expansion in Digital Libraries

Min Song
Center for Information Science and Technology, Temple University, Philadelphia, PA 19122
min.song@temple.edu

Il-Yeol Song
College of Information Science and Technology Drexel University Philadelphia, PA 19104
song@drexel.edu

Robert B. Allen
College of Information Science and Technology Drexel University Philadelphia, PA 19104
rba@drexel.edu

Zoran Obradovic
Center for Information Science and Technology, Temple University, Philadelphia, PA 19122
zoran@ist.temple.edu

ABSTRACT
In pseudo-relevance feedback, the two key factors affecting the retrieval performance most are the source from which expansion terms are generated and the method of ranking those expansion terms. In this paper, we present a novel unsupervised query expansion technique that utilizes keyphrases and POS phrase categorization. The keyphrases are extracted from the retrieved documents and weighted with an algorithm based on information gain and co-occurrence of phrases. The selected keyphrases are translated into Disjunctive Normal Form (DNF) based on the POS phrase categorization technique for better query refomulation. Furthermore, we study whether ontologies such as WordNet and MeSH improve the retrieval performance in conjunction with the keyphrases. We test our techniques on TREC 5, 6, and 7 as well as a MEDLINE collection. The experimental results show that the use of keyphrases with POS phrase categorization produces the best average precision.

Categories and Subject Descriptors
D.2.8 [**Information Storage and Retrieval**]: Information Search and Retrieval – Retrieval Models, Relevance Feedback, Query Formulation

General Terms
Algorithms, Design, Experimentation

Keywords
Information Gain, Keyphrase Extraction, Query Expansion, POS, WordNet

1. INTRODUCTION
In relevance feedback, relevance information is gathered from documents retrieved in a ranked list generated using an initial request. This relevance information is used to modify the search query and perform a further retrieval pass. The two main factors in relevance feedback are the source from which expansion terms are determined and the method of ranking expansion terms. These factors have a crucial impact on the retrieval performance in pseudo-relevance feedback. Pseudo-relevance feedback is an effective technique to retrieve more relevant documents without relevance feedback from users. In the pseudo-relevance feedback method, a small set of documents is retrieved using the original user-query. These documents, whose relevance is assumed, are then used to construct an expanded query, which is used, in turn, to retrieve the set of documents actually presented to the user.

In this paper, we present a novel unsupervised query expansion technique that utilizes keyphrases and Part of Speech (POS) phrase categorization. We use keyphrases extracted from the retrieved documents to improve term selection and query re-ranking for pseudo-relevance feedback. Keyphrase extraction is a process to extract important phrases in a document that the author or a cataloger would assign the document as keyword metadata [22]. Keyphrases are extracted from the top N-ranked documents retrieved and expansion terms selected from the keyphrase list rather than the whole document. The selected keyphrases are translated into Disjunctive Normal Form (DNF) by the POS phrase categorization technique.

We evaluate the keyphrases with the POS phrase categorization technique with TREC data. Retrieval results using TREC 5, 6, and 7 ad hoc tasks show that the use of keyphrases can improve pseudo-relevance feedback. Further, we explore a technique that combines synonymous terms from ontologies to keyphrases. However, there are mixed results using ontologies such as WordNet and MeSH for the query expansion task.

We also apply our technique to the biomedical domain where the task is to automatically discover the characteristics of documents that are useful for extraction of protein-protein interaction pairs, starting with only a handful of user-provided examples of instances of the relation to extract. We extract keyphrases from the retrieved documents from MEDLINE and use the top N ranked keyphrases for query expansion. Our technique retrieves more documents containing protein-protein pairs as the number of querying iterations increases.

This paper makes the following contributions. First, unlike most other query expansion techniques which use a single term selected with statistical-based term weighting, we use key phrases as the basic unit for our query term. The phrase selection relies on the overall similarity between the query concept and phrases of the collection rather than on the similarity between a query and the phrases of the collection [16]. We show that keyphrases extracted from the retrieved

JCDL'06, June 11–15, 2006, Chapel Hill, North Carolina, USA.

documents better represent the core concepts of the retrieved documents. Second, we propose a new notion of POS phrase categories, which is used to effectively combine multiple keyphrases into a disjunctive normnal form (DNF) used for query expansion. Third, our techniques can make use ontologies such as WordNet or MeSH to add more relevant phrases to the query. For WordNet, we employ a new word sense disambiguation technique. Our technique is novel in that it is based on the simility between senses in WordNet and keyphrases extracted from the retrieved documents. Fourth, we demonstrate that our proposed techniques are applicable to a variety of domains. We test our techniques on TREC data collections and biomedical data collections. Fifth, through extensive experiments, we validate the performance advantage of our techniques over other leading algorithms.The rest of the paper is organized as follows: Section 2 reviews the use of relevance feedback in ad hoc IR systems. Section 3 describes our keyphrase-based query expansion methods. Section 4 describes query expansion with ontologies. Section 5 outlines the test data. Section 6 reports on the experiments with TREC. Section 7 reports applying the technique to a MEDLINE database. Section 8 concludes the paper.

2. RELATED WORK

The quality of a query fed to an IR system has a direct impact on the success of the search outcome. In fact, one of the most important but frustrating tasks in IR is query formulation (e.g., [8]). Relevance feedback is a popular and widely accepted query reformulation strategy. The main idea consists of selecting important terms, or expressions, attached to the documents that have been identified as relevant by the user, and of enhancing the importance of these terms in a new query formulation. The expected effect is that the new query will be moved towards the relevant documents and away from the non-relevant ones.

As outlined in Section 1, pseudo-relevance feedback methods improve the retrieval performance on average but the results are not as good as relevance feedback. In pseudo-relevance feedback, problems arise when terms or phrases taken from assumed-to-be relevant documents that are actually non-relevant are added to the query causing a drift in the focus of the query. To tackle this issue, Mitra, et al. [15] incorporated term co-occurrences to estimate word correlation for refining the set of documents used in query expansion. They made use of only individual terms for query expansion whereas we utilize keyphrases for query expansion. In addition, they vary window sizes for matching queries but in our technique window sizes are determined by sentence lengths.

[14, 27] found that using the selected passages from documents for query expansion is effective in reducing the number of inappropriate feedback terms taken from non-relevant documents. Lam-Adesina and Jones [12] applied document summarization to query expansion. In their approach, only terms present in the summarized documents are considered for query expansion. Whereas Lam-Adesina and Jones add all terms from the summaries to the query, we use only the top N ranked keyphrases chosen with our selection criteria. Lam-Adesina and Jones adopted a summarization technique based on extracted summary sentences that are found by scoring the sentences in the documents. The scoring method is simply a sum of the scores gained by the four summarization methods: 1) Luhn's keyword cluster, 2) title terms frequency, 3) location/header, and 4) query-bias methods. Whereas their technique is based on simple mathematical properties of terms, our techniques are information theory-based as well as mathematically solid.

Liu et al. [13] used noun phrases for query expansion. Specifically, four types of noun phrases were identified: proper names, dictionary phrases, simple phrases, and complex phrases. A document has a phrase if all the content words are in the phrase within the defined window, and these documents that have matched phrases are considered to be relevant. They also apply a similarity measure to select the content words in the phrases which is positively correlated in the collection. By comparison, we utilize keyphrases including verb phrases from the top N ranked documents retrieved by the original query, whereas Liu et al. make use of only noun phrases in queries. In addition, our approach combines phrase co-occurrence with the Information Gain of a keyphrase.

Since we also investigate whether adding concepts from WordNet to keyphrases improves the retrieval performance, we briefly survey some related works to our approach. Liu et al. [13] add selected synonyms, hyponyms, and compound words based on their word sense disambiguation technique. Our approach to word sense disambiguation is different in that we disambiguate word sense by similarity criteria between all the non-stopwords from the synonyms and definitions of the hyponym synsets and keyphrases extracted from the retrieved documents. Voorhees' [27] used WordNet for adding synonyms of query terms whereas we use WordNet to add synonyms and substantial hyponyms of the top N ranked keyphrases.

3. KEYPHRASE-BASED PSEUDO-RELEVANCE FEEDBACK

We test whether carefully selected keyphrases can be effective for pseudo-relevance feedback. In this section, we discuss our techniques and procedures for query expansion. The following three subsections give detailed descriptions of the techniques used for keyphrase extracting, query re-weighting, and query translating used in our approach.

3.1 Keyphrase Extraction Procedures

Our keyphrase extraction procedure consists of two stages: 1) building an extraction model and 2) extracting keyphrases. The input of the "building extraction model" stage is training data. The input of the "extracting keyphrases" stage is test data or production data.

The keyphrases are extracted by referencing the keyphrase model. The keyphrase model for a target domain is learned by training sample inputs consisting of documents containing positive as well as negative examples. In our approach, the two keyphrase extraction stages are fully automated. Both training and test data are processed by the following three components: 1) Data Cleaning, 2) Data Tokenizing, and 3) Data Discretizing. Detailed descriptions are provided in the following subsections. These keyphrase extraction procedures have proven effective in other information extraction studies (e.g., [6, 23]).

3.1.1 Candidate Keyphrase Extraction Procedure

Input text is parsed into sentences. *Candidate keyphrases* are then selected within a sentence. The following three rules were used to select candidate keyphrases: 1) A keyphrase is limited to a certain maximum length. For this research, we set the maximum length at three consecutive words. 2) It cannot be a proper name (i.e., a single word that ever appears with an initial capital). 3) It cannot begin or end with a stop word. Ourstop word list from Okapi [29] consists of 256 unimportant terms. All continuous sequences of words in each document are evaluated as candidate phrases with these three rules.

3.1.2 Feature Selection

The following three feature sets were calculated for each candidate phrase: 1) Term Frequency * Inverse Document Frequency (TF*IDF), 2) Distance from First Occurrence (DFO), and 3) Part of speech (POS). TF*IDF is a well-established retrieval technique [20] for calculating the importance of a term in a document.

$$W_{ij} = tf_{ij} * \log_2 \frac{N}{n}$$

W_{ij} is the weight of term T_j in document D_i, and tf_{ij} is the frequency of term T_j in document D_i. N is the number of documents in a collection, and n is the number of documents where term T_j occurs at least once.

Distance from First Occurrence (DFO) is calculated as the number of phrases that precedes the phrase's first appearance, divided by the number of phrases in the document.

$$DFO = \sum w_{i-1} \big/ NP$$

w_{i-1} is the number of phrases preceding the target phrase, and NP is the total number of phrases in the document.

POS tagging assigns a POS such as noun, verb, pronoun, preposition, adverb, adjective or other lexical class marker to each word in a sentence. We combine the following four POS tagging techniques such as 1) NLParser, 2) Link-Grammar, 3) PCKimmo, and 4) Brill's tagger to improve POS tagging accuracy. This combined approach to POS techniques enables us to assign the best tag to lexical tokens, constituting candidate phrases by utilizing optimal features of each POS technique [23].

Because the features selected in our approach are continuous, we need to convert them into nominal forms to apply our machine learning algorithm. From many possible discretization algorithms, we chose equal-depth (frequency) partitioning which allows good data scaling [4]. Equal-depth discretization divides the range into N intervals, each containing approximately the same number of samples. The value of each feature, a candidate phrase, is replaced by the range to which the value belongs. Table 1 shows the results of discretization by an equal-depth partitioning. The values shown in Table 1 are derived from the TREC data.

Table 1: Discretization table.

Feature	*Discretization Range*				
	1	2	3	4	5
TF*IDF	< 0.003	>= 0.003 && < 0.015	>= 0.015 && < 0.050	>= 0.050 && < 0.100	>= 0.100
DFO	< 0.150	>= 0.150 && < 0.350	>=0.350 && < 0.500	>= 0.500 && < 0.700	>= 0.700
POS	< 0.001	>= 0.001 && < 0.200	>=0.200 && < 0.700		

3.2 Keyphrase Ranking

Automatic query expansion requires a term-selection stage. The ranked order of terms is of primary importance in that the terms that are most likely to be useful are close to the top of the list. We re-weight candidate keyphrases with Information Gain. Specifically, candidate keyphrases are ranked by an Information Gain, *GAIN(P),* measure of expected reduction in entropy based on the "usefulness" of an attribute *A*. This is one of the most popular measures of associations used in data mining. For instance, Quinlan [17] uses Information Gain for ID3 and its successor C4.5 which are widely-used decision tree techniques. ID3 and C4.5 construct simple trees by choosing at each step the splitting feature that "tells us the most" about the training data. Mathematically, Information Gain is defined as.

$$GAIN(P_i) = I(p,n) - E(P_i)$$

Where P_i is value of a candidate phrase that falls into a discretized range. *I(p,n)* measures the information required to classify an arbitrary tuple.

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

S, is an example set that contains S_i tuples of class C_i for *i=(1,...,m).*

$$E(W) = \sum_{i=1}^{m} - P_i \log_2 P_i$$

where Pi is the proportion of W belonging to class *i*. Note that the target attribute takes on *m* possible values, and the maximum possible entropy is $\log_2 m$. Each candidate phrase, extracted from a document, is ranked by the probability calculated with *GAIN(P)*. In our approach, *I(p,n)* is stated such that class p:

candidate phrase = "keyphrase" and class n: candidate phrase = "non-keyphrase."

Example. Suppose that a candidate phrase, "text mining algorithm", has 0.0134 for TF*IDF feature. Thus, according to Table 1, it falls into Range 2 for TF*IDF. Let us also assume that it appears twice as a keyphrase in the range, tf_idf_3, the third range for tf*idf in Table 1. In the given scenario, the number of tuples in tf_idf_3 is 20. The number of keyphrases is 120 out of the total 1,000 candidate phrases. For this case, *I(120,880)* is 0.529361, *E(tf_idf_3)* is 0.046900, and *GAIN(Ptf_idf_3)* = 0.482462. *GAIN(P)* for DFO and POS is calculated in the same way. The overall rank of a candidate phrase is determined by the sum of $GAIN(P_i)$. In the above example, the candidate phrase is ranked fifth as a keyphrase for the given test document.

Many query re-ranking algorithms are reported in the literature [18, 21]. These algorithms attempt to quantify the value of candidate query expansion terms. Formulae estimate the term value based on qualitative or quantitative criteria. The qualitative arguments are concerned with the value of the particular term in retrieval. On the other hand, the quantitative argument involves some specific criteria such as a proof of performance. One example of the qualitative-based formula is the relevance weighting theory.

While there are many promising alternatives to this weighting scheme in the IR literature [28], we chose the Robertson-Sparck Jones algorithm [19] as our base because it has been demonstrated to perform well and is naturally well suited to our task. In addition, incorporating other term weighting schemes does not require changes to our system architecture.

Robertson and Sparck Jones proposed the F4.5 formula. It has been widely used in IR systems such as Okapi with some modifications. Although a few more algorithms were derived from F4.5 formula by Robertson and Spark Jones, in this paper, we modify the original formula for keyphrases as follows:

$$P(w) = \log \frac{\left(\frac{r + 0.5}{R - r + 0.5}\right)}{\left(\frac{n - r + 0.5}{N - n - R + r + 0.5}\right)}$$

P(w) is keyphrase weight, *N* is the total number of sentences, *n* is the number of sentences in which that query terms co-occur, *R* is the total number of relevant sentences, and *r* is the number of relevant sentences in which the query terms co-occur.

We combine Information Gain with the modified F4.5 formula to incorporate keyphrase properties gained as follows:

$$KP(r) = \sqrt{\frac{GAIN(p) * P(w)}{2}}$$

All candidate keyphrases are re-weighted by *KP(r)* and the top *N* ranked keyphrases are added to the query for the next pass. The *N* number is determined by the size of the retrieved documents. KP(r) accounts for both the information gain value and the query re-weighting value of candidate phrases.

3.3 Query Translation into DNF Using POS Phrase Categorization

A major research issue in IR is how to ease the user's role of query formulation through automating the process of query formulation. There are two essential problems to address when searching with online systems: 1) initial query formulation that expresses the user's information need; and 2) query reformulation that constructs a new query from the results of a prior query [19]. The latter effort implements the notion of relevance feedback in IR systems and is the topic of this section.

An algorithm for automating Boolean query formulation was first proposed in 1970. This method employs a term weighting function first described in Frants et al. [7] to decide the "importance" of terms which have been identified. The terms were then aggregated into "sub-requests" and combined into a Boolean expression in disjunctive normal form (DNF). Other algorithms that have been proposed to translate a query to DNF are based on classification [8], decision-trees [3], and thesauri [24]. Hearst [10] proposed a technique for constructing Boolean constraints, which was revisited by Mitra et al. [15].

Our POS category-based translation technique differs from others in that ours is unsupervised and is easily integrated into other domains. In our technique, there are **four different phrase categories** defined; 1) Ontology phrase category, 2) Non-Ontology noun phrase category, 3) Non-Ontology proper noun phrase category, and 4) Verb phrase category. Phrases that have corresponding entities in ontologies such as WordNet or MeSH belong to the ontology phrase category. We include the Verb phrase category as a major category because important verb phrases play a role in improving the retrieval performance [9].

Figure 1: Sample keyphrases extracted for query expansion.

```
<keyphrases id="350">
   <keyphrase weight="0.39282" category="2" >
         computer screen </keyphrase>
   <keyphrase weight="0.38114" category="1" >
         occupational health </keyphrase>
   <keyphrase weight="0.38566" category="1" >
         workplace disorders </keyphrase>
   <keyphrase weight="0.38432" category="1" >
         physical injury</keyphrase>
   <keyphrase weight="0.38427" category="1" >
         computer terminal </keyphrase>
   <keyphrase weight="0.38320" category="2" >
         workers computer </keyphrase>
  <keyphrase weight="0.38293" category="4">
         report </keyphrase>
   <keyphrase weight="0.38174" category="1" >
         terminals activity </keyphrase>
</keyphrases>
```

Keyphrases within the same category are associated with a facet of the concept; therefore, the keyphrases within the category are translated into DNF, which each keyphrase is OR-ed together. The inter-categories are then translated into Conjunctive Normal Form, which is AND-ed together with Boolean operator 'AND'.

The sample keyphrases for Query 350 for TREC-6 are shown in Figure 1. As explained earlier, within the same category the phrases are combined with the OR Boolean operator. Between categories, the terms are combined with the AND Boolean operator. Thus, the query shown in Figure 1 is translated as follows:

((occupational health OR workplace disorders OR physical injury OR computer terminal OR terminals activity) AND (computer screen OR workers computer) AND report).

4. QUERY EXPANSION WITH ONTOLOGIES

For the top N ranked keyphrases, our technique can traverse ontologies such as WordNet or MeSH. If a phrase appears in the ontology, the keyphrase is categorized as an Ontology phrase category. With WordNet, we encounter a complication with multiple senses of a given phrase.

To tackle this problem, we introduce a straightforward Word sense disambiguation technique, which is based on similarities between WordNet phrases and the keyphrases extracted by our technique. In WordNet, a group of synonyms with the same meaning composes a "synset". The synsets are linked to each other through relationships such as hyponyms, hypernyms, and holonyms. If no synsets are found for the given phrase, we traverse down in the synset list to find the next sysnet related to the input phrase. For multiple synsets, all the non-stopwords are captured from synonyms and their descriptions, hyponyms and their descriptions, and other relations for each synset. These terms and phrases are then compared with the keyphrase list by the similarity function *Sim(S)*. Our word disambiguation technique is based on the topical relevance between senses and keyphrases extracted from the documents.

$$Sim(S) = \sum_{i=1}^{M} \max_{j \in \{1,..,n_i\}} w(p_{ij})$$

where $w(p_{ij})$ is the frequency of phrase p_{ij} if it occurs in a synset, S, and is 0 otherwise. The synset with the highest similarity value is chosen and synonyms from the synset are added for query expansion.

5. TREC TEST DATASETS

The keyphrase-based query expansion method is evaluated using the TREC-5, TREC-6, and TREC-7 ad hoc test sets. The ad hoc task investigates the performance of systems that search a static document collection using new query statements. The document set consists of approximately 628,531 documents distributed on three CD-ROM disks (TREC disks 2, 4, and 5) taken from the following sources: Federal Register (FR), Financial Times (FT), Foreign Broadcast Information Service (FBIS), LA Times (LAT), Wall Street Journal, AP Newswire, and Information from Computer Select disks.

Search requests in the form of TREC topics consist of three parts: title, description, and narrative. The title consists of individual words that best describe the information need, the description field is a one-sentence description of the topic area, while the narrative gives a concise description of what makes a document relevant or not. The different parts of the TREC topic allow investigation of the effect of different query lengths on retrieval performance. For our investigation only the title field of the topics is used because it is most similar to the form of queries entered by typical users.

Table 2: Documents and queries used in TREC ad hoc tasks.

Task	Documents	Queries
TREC5	TREC disks 2,4	251-300
TREC6	TREC disks 4,5	301-350
TREC7	TREC disks 4,5	351-400

The query sets and document collections used in these tasks are shown in Table 2. We use ZETTAIR [2] as the underlying IR system. It is easy to add a query expansion technique such as ours on top of it. In addition, Billerbeck and Jobel [2] reported that ZETTAIR produced comparable results with Okapi.

6. EVALUATION OF QUERY EXPANSION FOR TREC DATASETS

Most of the top performing query expansion techniques use a term weighting method developed in either the Okipi system or the SMART system [22]. In our experiments with TREC datasets, we employ BM25, an Okapi formula for evaluation. We also employ a machine learning technique, called SLIPPER [4], Adaboost-based query expansion technique. Adaboost algorithms were used for query expansion in the Information Extraction (IE) tasks and proved to be an effective QE technique [1]. The other three algorithms are based on our QE techniques.

The algorithms used in the experiments are denoted as follows:

BM25: The standard Okapi BM25 formula is used as the baseline:

$$BM25 = \sum_{t \in q} \log\left(\frac{N - f_t + 0.5}{f_t + 0.5}\right) \times \frac{(k_i + 1) f_{d,t}}{K + f_{d,t}}$$

where t is a term of query q. f_t is the number of occurrences of a particular term across the document collection that contains N documents and $f_{d,t}$ is the frequency of a particular term t in document d. K is $k_1((1-b)+b * L_d)/ AL$, where k_1 and b are parameters set to 1.2 and 0.75, respectively. L_d is the length of a particular document and AL is the average document length.

SLP: SLP (SLIPPER) is an efficient rule-learning system, which is based on confidence-ruled boosting, a variant of AdaBoost [4]. SLIIPPER learns concise rules such as ***"protein AND interacts" --> Useful***, which shows that if a document contains both term *protein* and term *interacts*, it is declared to be useful. These classification rules generated by SLP are then translated into conjunctive queries in the search engine syntax. For instance, the above rule is translated into a query "protein AND interacts."

KP: Apply the Keyphrasebased query expansion algorithm described in Section 3.

KP+C: In addition to the KP formula, this algorithm employs Boolean constraints by POS type of keyphrases.

KP+C+O: In addition to KP+C, this algorithm employs Ontologies as outlined in Section 4.

Table 3: Results for TREC 5 with our five query expansion algorithms executing the query set 251-300.

Algorithm	TREC 5	
	Avg. P	P@20
BM25	0.1623	0.3252
SLP	0.1299	0.2656
KP	0.1938	0.3368
KP+C	0.1985	0.3398
KP+C+O	0.2012	0.3371

Table 3 shows the overall performance of the five algorithms executing the query set 251-300 on TREC 5 data. The results show that KP+C+O has the best performance in average precision as well as in precision at top twenty ranks (P@20) compared to other algorithms. The exception is that KP+C in P@20 shows the best improvement among the algorithms.

To confirm the differences among the conditions, we conducted an ANOVA for the P@20 TREC 5 results. This showed an overall effect of condition $F(3,196)=17.64$, $p<0.01$. We also conducted individual *t*-tests essentially as specific comparisons. Our prediction that KP would be better than BM25 was confirmed $t(49)=-7.37$, $p<0.01$ (one-tailed) at n-1 degrees of freedom (50 queries). Similarly, our prediction that KP+C would be better than KP was confirmed $t(49)=-4.72$, $p<0.01$ (one-tailed). However, our original expectation that KP+C+O would be better than KP+C was not confirmed $t(49)=1.98$ and was, in fact, in the wrong direction.

Table 4: Results for TREC 6 with our five query expansion algorithms.executing the query set 301-350

Algorithm	TREC 6	
	Avg. P	P@20
BM25	0.1797	0.3160
SLP	0.1358	0.2654
KP	0.2098	0.3390
KP+C	0.2114	0.3424
KP+C+O	0.2053	0.3410

Tables 4 and 5 show similar results to those obtained for TREC 5. The three new algorithms improve the retrieval performance on TREC 6 and 7. As with TREC 5, the KP+C algorithm outperforms BM25, SLP, and KP+C+O algorithms in average precision and in P@20.

Table 5: Results for TREC 7 with our five query-expansion algorithms executing the query set 351-400.

Algorithm	TREC 7	
	Avg. P	P@20
BM25	0.2229	0.3837
SLP	0.1502	0.3044
KP	0.2343	0.3878
KP+C	0.2458	0.4024
KP+C+O	0.2319	0.3985

Our keyphrase-based technique combined with the POS phrase category produces the highest average precision. One of the best results on TREC 5 is 19.44 and 32.40 in average precision and P@20 respectively [15]. On TREC 6, their best results are 20.34 and 33.50 in average precision and P@20. The algorithm KP+C produces 21% and 48% better than these results on TREC 5 in average precision and P@20. On TREC 6, it is 39% and 22% which are better than the results reported in [15].

7. EXPERIMENTS ON MEDLINE DATASETS

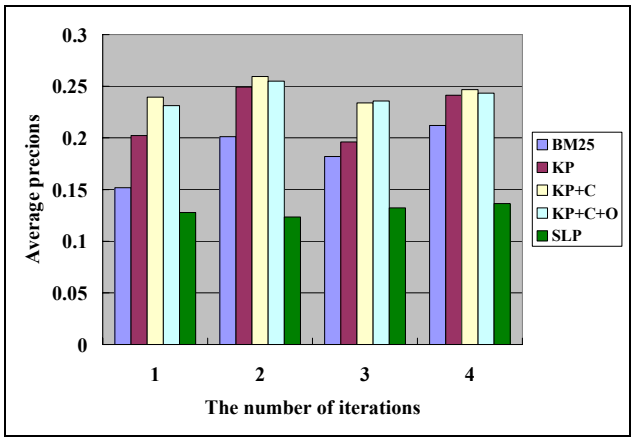
To explore the flexibility and generality of our algorithms, we explored query expansion for MEDILINE articles. The task we selected is to retrieve documents containing protein-protein interaction pairs. The data sets are composed of abstracts collected from the MEDLINE database. MEDLINE contains more than 12 million documents. In order to measure the accuracy rate, we count the number of documents retrieved from MEDLINE that contain protein-protein pairs. The protein names are collected from the Database of Interacting Proteins (DIP) and Protein-Protein Interaction Database (PPID) databases. For protein-protein interaction tasks, we use PUBMED as the underlying IR engine and MeSH as an ontology. Initial queries consist of 3 to 5 protein-protein interaction pairs. Figure 2 shows the initial query used to retrieve the documents from PUBMED.

Figure 2: Initial query used for protein-protein interaction tasks.

```
<init_query>
   <terms protein1="MAP4" protein2="Mapmodulin"/>
   <terms protein1="WIP" protein2="NCK"/>
   <terms protein1="GHR" protein2="SHB"/>
   <terms protein1="SHIP" protein2="DOK"/>
   <terms protein1="LNK" protein2="GRB2"/>
   <terms protein1="CRP" protein2="Zyxin"/>
</init_query>
```

The experimental results for MEDLINE are shown in Figure 3. Our three algorithms improve the performance in retrieving documents containing protein-protein interaction pairs, compared with BM25 and SLP. As with our TREC results, the KP+C algorithm gives the best average precision.

Figure 3: Experimental results for MEDLINE with our five query expansion algorithms



We also explored the effect of a sequence of query expansion iterations. Table 6 shows the results for five query expansion iterations. The second column is the number of retrieved documents from MEDLINE for each iteration. The third column shows the number of retrieved documents containing protein-protein pairs. The fourth column is the F-Measure [25]. In the F-Measure, we use *b=2* since because recall is more important than precision in the tasks of retrieving the documents containing protein-protein interaction pairs. Our results show that the F-Measure generally increases as the number of iterations increases.

Table 6: Query expansion iterations for MEDLINE.

Iteration	No of retrieved documents	No of documents containing protein-protein pairs	F-Measure (%)
1	30	18	47.76
2	609	289	51.65
3	832	352	51.27
4	1549	578	53.69

7. CONCLUSION

In this paper, we presented an effective unsupervised query expansion technique based on keyphrases and the POS phrase categories. Encouraged by the previous studies on pseudo-relevance feedback we applied keyphrase extraction techniques to query expansion. Along with keyphrase-based expansion, we employed a similarity-based word sense disambiguation technique in using an ontology (e.g., WordNet) to add terms to the query. We also employed a POS phrase category-based Boolean constraint technique to combine multiple phrases into a single expanced query.

We demonstrated that our techniques yield significant improvements over the well-established BM25 and Adaboost algorithms for the three TREC collections, TREC 5, 6, and 7, as well as for MEDLINE data on the protein-protein interaction tasks. Among five algorithms implemented, BM25, KP, KP+C, and KP+C+O, the KP+C algorithm seems to be the best. The reason that the KP+C+O is not superior to the KP+C as hypothesized might be because these ontologies are applied to already enriched keyphrases.

Our paper makes the following contributions. First, unlike most other query expansion techniques, we use key phrases as the basic unit for our query term. We have shown that keyphrases extracted from the retrieved documents better represent the core concepts of the retrieved documents. Second, we presented a new query reformulation technique based on POS phrase categorization to combine the phrases into a Disjunctive Normal Form. Third, we have shown that our techniques can make use of an ontology such as WordNet or MeSH to add more relevant phrases to the query. For WordNet, we employed a new word sense disambiguation technique, which is based on the similarity between senses and keyphrases extracted from the retrieved documents. Fourth, we have shown that the techniques are applicable to a variety of domains. We test our techniques on TREC data collections and biomedical data collections. We have shown that the experiments show the promising results on both. Fifth, through extensive experiments, we have validated the performance advantage of our techniques over other leading algorithms.

In the future work, we will employ a more fine-tuned word sense disambiguation technique such as [13] to improve the retrieval accuracy with WordNet further. In our earlier work [11] we applied a medical ontology, Unified Medical Language System (UMLS), to entity tagging with some promising results. As a follow-up study, we are investigating whether UMLS improves the retrieval accuracy of the documents containing protein-protein pairs.We are also interested in whether keyphrases help the users understand the content of a collection and provide sensible entry points into it. In addition, we will investigate whether and how keyphrases can be used in information retrieval systems as descriptions of the documents returned by a query, the basis for search indexes, a way of browsing a collection, and a document clustering technique.

8. ACKNOWLEDGMENTS

Computing support for this project was facilitated by IBM SUR grant.

9. REFERENCES

[1] Agichtein, E. and Gravano, L. (2003). Querying text databases for efficient information extraction. In *Proceedings of the 19th IEEE International Conference on Data Engineering*, 113-124.

[2] Billerbeck, B., and Zobel, J. (2004). Questioning Query Expansion: an Examination of Behavior and Parameters, in: *Proceedings of Fifteenth Australasian Database Conference*: 69-76.

[3] Chang, K.C., Garcia-Molina, H., and Paepcke, A. (1996). Boolean Query Mapping Across Heterogeneous Information Sources, *IEEE Transactions on Knowledge and Data Engineering*, 8(4): 515-521.

[4] Cohen, W.W., and Singer, Y. (1999). Simple, Fast, and Effective Rule Learner, In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence and*

Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22: 335-342.

[5] Dougherty, J., Kohavi, R., and Sahami, M. (1995) Supervised and Unsupervised Discretization of Continuous Features. In: *Proceedings of ICML-95, 12th International Conference on Machine Learning*, Lake Tahoe, US: 194-202.

[6] Frank E., Paynter G.W., Witten I.H., Gutwin C., and Nevill-Manning, C.G. (1999) Domain-specific Keyphrase Extraction, In: *Proceedings of Sixteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA: 668-673.

[7] Frants, V.I., and Shapiro, J. (1991). Algorithm for Automatic Construction of Query Formulations in Boolean Form, *JASIS*, 42(1): 16-26.

[8] French, J.C., Brown, D.E., and Kim, N.H. (1997). A Classification Approach to Boolean Query Reformulation, *Journal of the American Society for Information Science*, 48(8): 694-706.

[9] Gauch, S., Wang, J., and Rachakonda, S.M. (1997). A Corpus Analysis Approach for Automatic Query Expansion and its Expansion to Multiple Databases, *ACM Transaction on Information Systems*, 17: 250-269.

[10] Hearst, M.A. (1996). Improving Full-Text Precision on Short Queries Using Simple Constraints, In: *Proceedings of the Symposium on Document Analysis and Information Retrieval*.

[11] Hu, X., Lin, T.Y., Song, I-Y, Lin, X, Yoo, I., and Song. M. (2004). An Ontology-based Scalable and Portable Information Extraction System to Extract Biological Knowledge from a Huge Collection of Biomedical Web Documents, In: *Proceedings of the 2004 IEEE/ACM Web Intelligence Conference*: 77-83.

[12] Lam-Adesina A.M., and Jones, G.J.F. (2001). Applying Summarization Techniques for Term Selection in Relevance Feedback, *Proceedings of the 24th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*: 1-9.

[13] Liu, S., Liu, F., Yu, C., and Meng, W. (2004). An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases, *Proceedings of the 27th annual international Conference on Research and development in Information Retrieval*: 266-272.

[14] Mihalcea, R., and Moldovan, D. (2000). Semantic Indexing Using WordNet Senses. ACL *Workshop on IR & NLP*.

[15] Mitra, C.U., Singhal, A., and Buckely, C. (1998). Improving Automatic Query Expansion, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*: 206-214.

[16] Qiu, Y., and Frei, H. (1993). Concept-based Query Expansion, In: *Proceedings of the 16th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*: 160-169.

[17] Quinlan, J. R. (1993). *Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann Publishers.

[18] Ro, J.S. (1988). Evaluation of the Applicability of Ranking Algorithms, Pt. I and Pt. II. *Journal of the American Society for Information Science*. 39; 73-78: 147-160.

[19] Robertson, S.E., and Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, (27): 129-146.

[20] Salton, G., Buckley, C., and Fox, E.A. (1983). Automatic query formulations in information retrieval. *Journal of the American Society for Information Science*, 34(4):262-280, July 1983.

[21] Sager, W.K.H., and Lockemann, P.C. (1976). Classification of Ranking Algorithms. *International Forum for Information and Documentation.* 1:12-25.

[22] Singhal, A. and Kaszkiel, M. (2001). A case study in web search using TREC algorithms, Proceedings of the 10th international conference on World Wide Web, 708-716.

[23] Song, M., Song, I.-Y., and Hu, X. (2004). Designing and Developing an Automatic Interactive Keyphrase Extraction System with UML, *ASIST Annual Meeting*, Providence, RI: 367-372.

[24] Van Der Pol, R. (2003). Dipe-D: a Tool For Knowledge-based Query Formulation, *Information Retrieval,* 6:21-47.

[25] Van Rijsbergen, C.J. (1979). *Information Retrieval*. Butterworth, London.

[26] Voorhees, E.M. (1994). Query Expansion Using Lexical-Semantic Relation, Proceedings of 17th International Conference Research and Development in Information Retrieval, pp. 61-69, 1994.

[27] Voorhees, E.M. (1998). Using WordNet for Text Retrieval. In *WordNet, an Electronic Lexical Database*, C. Fellbaum (ed.), MIT Press, 285-303

[28] Xu, J., and Croft, W.B. (2000). Improving the Effectiveness of Information Retrieval with Local Context Analysis, *ACM Transactions on Information Systems*, 18(1): 79-112.

[29] Okapi. http://www.soi.city.ac.uk/~andym/OKAPI-PACK/