College of Information Science and Technology

Integration of Association Rules and Ontology for Semantic-based Query Expansion

Min Song [a,*], Il-Yeol Song [b], Xiaohua Hu [b], Robert B. Allen [b]

[a] Department of Information Systems, New Jersey Institute of Technology, University Heights Newark, NJ 07102 USA

[b] College of Information Science & Technology, Drexel University, Philadelphia, PA, 19104 USA

Abstract. The goal of query expansion is to reduce the mismatch between documents and queries by expanding the query using words or phrases with a similar meaning or some other statistical relation to the set of relevant documents. One of the limitations with query expansion techniques is that a query is often expanded only by the linguistic features of terms. To tackle this problem, we propose a novel semantic query expansion technique that combines association rules with ontologies and Natural Language Processing techniques. Our technique utilizes the association rule discovery to find good candidate terms to improve the retrieval performance. These candidate terms are automatically derived from collections and added to the original query. Our technique is differentiated from others in that 1) it utilizes the semantics as well as linguistic properties of unstructured text corpus, 2) it makes use of contextual properties of important terms discovered by association rules, and 3) ontologies' entry is added to the query by disambiguating word senses. Experiments conducted on TREC collections give encouraging results. We achieve from 13.41% to 32.39% improvement in term of P@20 and from 8.39% to 14.22% in terms of F-measure with TREC ad hoc queries. Detailed descriptions of the experimental results are discussed in the paper.

1 Introduction

An Information Retrieval (IR) System consists of a database, containing a number of documents, an index, that associates each document to its related terms, and a matching mechanism, that maps the user's query, consisting of terms, to a set of

associated documents. A typical goal of an IR system is to find a set of documents containing information needed by searchers in the given indexed database(s). In processing queries that searchers formulate, the conventional IR query languages require the searcher to state precisely what they want. Searchers need to be able to express their needs in terms of precise queries (either in Boolean form or natural languages). However, due to searchers' lack of knowledge in the search domain (anomalous state of knowledge -- An anomaly in one's state of knowledge, or lack of knowledge, with respect to a problem faced), a query syntax formulated by searchers often does not meet the searchers' information needs. In addition, a single-term-query that a normal user formulates often retrieves many irrelevant articles as well as fails to find hidden knowledge or relationships buried in content of the articles.

To overcome this limitation with query formulation, many IR systems provide facilities for relevance feedback, with which searchers can identify documents of interest to them. IR systems can then use the keywords assigned to these desired documents to find other potentially relevant documents. However, these IR systems fail to distinguish among the attributes of the desired documents for their relative importance to the searchers' needs.

With these issues in current Query Expansion (QE) techniques in mind, we introduce a novel querying technique, called SemanQE, combining association rules, ontologies, and IR techniques to retrieve promising documents for information retrieval. SemanQE has several unique strengths over other QE techniques. First, it proposes a hybrid query expansion algorithm combining association rules with ontologies and natural language processing techniques. Second, our technique utilizes the semantic as well as linguistic properties of unstructured text corpus. Third, our technique makes use of contextual properties of important terms discovered by association rules. To evaluate the performance of SemanQE, we compare SemanQE with cosine similarity-based QE, Okapi BM25[11], and SLIPPER[2]. Okapi BM25 is a powerful probabilistic query expansion technique widely used in IR and SLIPPER is a rule-based AdaBoost technique. We also investigate whether ontologies impact retrieval performance.

This paper makes the following contributions: (1) This method utilizes the semantic, as well as linguistic, properties of unstructured text corpus and thus our system is able to expand queries based on indirect associations embedded among the terms. (2) Our method uses of contextual properties of important terms discovered by association rules. (3) With similarity-based word sense disambiguation technique, ontologies' entries are added to the query set. (4) We demonstrate the effectiveness of our method through experiments conducted on a subset of TREC collections. We achieve from 13.41% to 16.93% improvement in term of P@20 with TREC-5 ad hoc queries. With TREC-6 and TREC-7 ad hoc queries, we observe from 24.18% to 32.39% improvements and from 17.85% to 21.51% respectively in terms of P@20. In terms of F-measure, we achieve from 8.39% to 14.22% in terms of F-measure with TREC-5, TREC-6, and TREC-7 ad hoc queries.

The remainder of paper consists of the following chapters: Section 2 summarizes the related work. Section 3 describes the overall architecture of SemanQE. Section 4 describes the evaluation. Section 5 concludes the paper.

2 Related Work

The quality of a query for 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 [3]. 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.

Pseudo-relevance feedback methods improve 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. [9] incorporated term co-occurrences to estimate word correlation for refining the set of documents used in query expansion.

Mihalcea and Moldovan [8] 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 [5] applied document summarization to query expansion. In their approach, only terms present in the summarized documents are considered for query expansion. Lam-Adesina and Jones adopted a summarization technique based on sentence-extracted summaries 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. [6] 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 to be positively correlated in the collection.

Latiri et al. [7] approached query expansion by considering the term-document relation as fuzzy binary relations. Their approach to extract fuzzy association rules is based on the closure of an extended fuzzy Galois connection, using different semantics of term membership degrees.

Because we also investigate whether adding concepts from WordNet to query sets by SemanQE improves the retrieval performance, we briefly survey some related works to our approach. Liu et al. [6] 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' [14] used WordNet for adding synonyms of query terms whereas we use WordNet to add synonyms and substantial hyponyms of the top N ranked terms and phrases.

3 The Semantic Query Expansion System

In this section, we describe the semantic query expansion system. In Section 3.1, we present the system architecture of our semantic query expansion system. In Section 3.2, we discuss the ontology used in our method. Finally, Section 3 explains our semantic query expansion algorithm called SematicQE.

3.1 The System Architecture

The system architecture of our semantic query expansion system, SemanQE, is illustrated in Fig. 1. SemanQE consists of three major components: 1) core association rule-based query expansion 2) feature selection, and 3) ontologies-based expansion components.

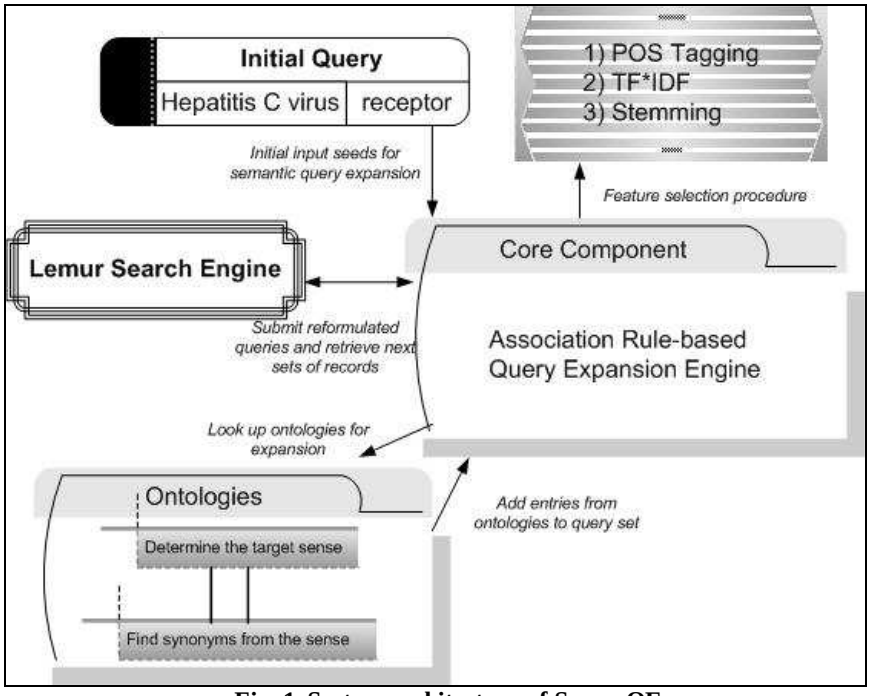

Fig. 1. System architecture of SemanQE

We use the Lemur IR system as a backend engine for SemanQE in that Lemur is robust and achieves high accuracy in terms of precision[10]. Lemur is developed by collaboration between the Computer Science Department at the University of Massachusetts and the School of Computer Science at Carnegie Mellon University. Lemur is designed to facilitate research in language modeling and information retrieval. The core association rule-based expansion algorithm is based on a well-known Apiori algorithm [1]. The apriori algorithm has been widely used to mine useful knowledge in large transaction databases. The support of a set of items in a

transaction database is the fraction of all transactions containing the itemset. An itemset is called frequent if its support is greater or equal to a user-specified support threshold. An association rule is an implication of the form X => Y where X and Y are disjoint itemsets. To apply association rule mining to our query expansion, we assume that each document can be seen as a transaction while each separate word inside can also be seen as items, represented by wordset.

The feature selection component processes the input documents to select important terms. In doing so, unimportant words such as functional words and stop words are excluded. We applied TF*IDF technique to extract important terms and phrases. In addition, we applied a POS tagging technique to filter out less important terms in terms of POS tags. TF*IDF was first proposed by Salton and Buckley [13]. It is a measure of importance of term in a document or class. Brill POS Tagger is chosen for our POS tagger. Brill's technique is one of the high quality POS tagging techniques.

Ontologies component expands queries selected from the core component. WordNet is used as ontologies for our system. With a set of terms and phrases, we first disambiguate word senses based on formula proposed in Section 3.2. WordNet is then referenced to find relevant entries semantically and syntactically.

The outline of the approach described in Figure 1 is as follows:

Step 1: Starting with a set of user-provided seed instances (the seed instance can be quite small), our system retrieves a sample of documents from the backend indexes via a search engine. At the initial stage of the overall document retrieval process, we have no information about the documents that might be useful for extraction. The only information we require about the target answer sets is a set of user-provided seed instances. We use some simple queries (just use the attribute values of the initial seed instances) to extract the document sample of pre-defined size from the search engine.

Step 2: On the retrieved document set, we parse each document into sentences and apply IR and natural language processing techniques to select important terms and phrases from the input documents.

Step 3: Applying a hybrid querying expansion algorithm that combines association rules and ontologies to derive queries targeted to match and retrieve additional documents similar to the positive examples.

Step 4: Reformulate queries based on the results of Step 3 and query the search engine again to retrieve the improved result sets matched to the initial queries.

Fig. 2 shows the how SemanQE works and what output it generates in each step.

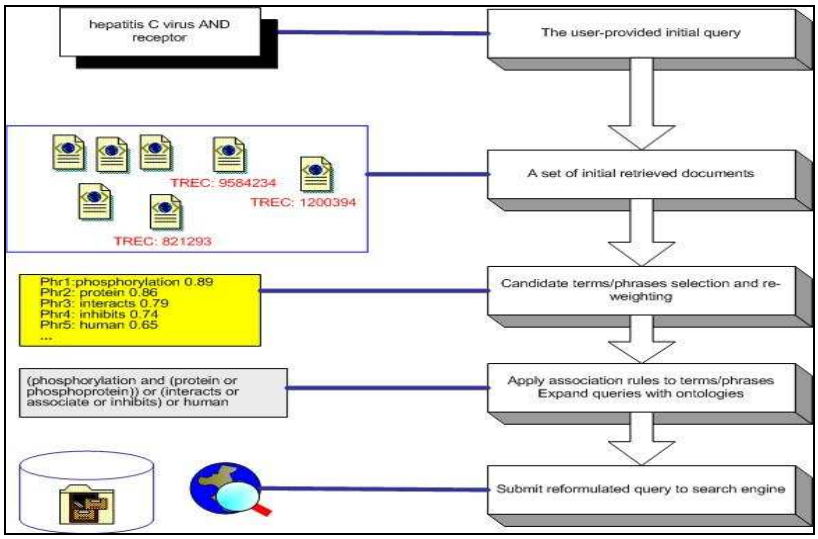

Fig 2: Procedures of the System Workflow

3.2 Ontologies

We adopted WordNet for ontologies of SemanQE. WordNet is an online lexical reference system in which English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets in WordNet. Early results on sense-based query expansions were not encouraging [12, 13]. However, more recent work [4] analyzes the effect of expanding a query with WordNet synsets, in a "canned" experiment where all words are manually disambiguated. Our usage of WordNet is for retrieving the promising documents by expanding queries syntactically and semantically. We traverse WordNet hierarchy to find out the best entries for the terms to be expanded.

A challenging problem with WordNet that we encounter is that there are multiple senses of given a term. 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 sysnet. 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).*

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

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.

3.3 The SemanQE Algorithm

In this section, we provide details of SemanQE algorithm. As shown in Table 1,

Table 1: Association Rule-based SemanQE Algorithm

(1) Retrieve initial results from Lemur based on the queries provided by a user
(2) Select important noun and phrases from RD (retrieved documents) by POS ad TF*IDF
(3) Apply Apriori to find all X->Y rules
For (i = 1; i < Size of L_i; i++) do
(4) Build Qi based on rules generated by Step 2.
(5) Apply Ontologies to expand Qi.
(6) Query Lemur with Q_i constructed by Step 5
If hit count != 0
(7) Retrieve TREC records for information extraction

SemanQE takes the user-provided queries to retrieve the initial set of documents from Lemur. The general description of the algorithm is as follows: Once the data were parsed, the important nouns and noun phrases were extracted based on the following two techniques: TF*IDF and Brill's Part of Speech (POS) tagging technique [2]. After the important noun and noun phrases are extracted, the Apriori algorithm [1] is applied andSemanQE builds a set of queries based. Finally, we applied ontologies to expand queries generated by association rules. As example of the query is:
(Adult+AND+Antineoplastic+Combined+Chemotherapy+Protocols+AND+Dacarbazine)+NOT+raynaud

Lemur was then searched with the query constructed by Step 4 and retrieve TREC records. In the feature selection, important terms or phrases are represented in the following term x document matrix.

$$D_i = t_{i1}, t_{i2}, ..., t_{im} \quad (1)$$

Each document in the retrieved results (D_i) consists of vector of selected terms or phrases (t_{im}). The terms and phrases that exceed the threshold are included in the vector as the input for semantic association rules.

4 Evaluation

We present the data collections used for the experiments, the experimental methods, and the other QE techniques for comparison. To evaluate SemanQE, we compare it with two other query expansion techniques: 1) Cosine similarity-based, a traditional IR technique for the vector space model, 2) SLIPPER, a rule-based query expansion, and 3) Okapi BM25, a probabilistic query expansion. Performance of these techniques is measured by F-measure and P@20. The data used for experiments are retrieved from TREC via the Lemur search engine.

4.1 Data Collection

The Text Retrieval Conference (TREC) is sponsored by both the National Institute of Standards and Technology (NIST) and U.S. Department of Defense. NIST TREC Document Databases (TREC data) are distributed for the development and testing of IR systems and related natural language processing research. The document collections consist of the full text of various newspaper and newswire articles plus government proceedings. The documents have been used to develop a series of large IR test collections known as the TREC collections.

Our 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), Los Angeles Times (LAT), Wall Street Journal, AP Newswire, and Information from Computer Select disks.

The format of the documents on the TREC disks is a labeled bracketing expressed with XML tags. The datasets on the disks have identical major structures but have different local structures. Every document is bracketed by <DOC>…</DOC> tags and has a unique document identifier, bracketed by <DOCNO>…</DOCNO> tags.

Tables 2, 3, and 4 show the statistics of records contained in three disks respectively.

Table 2. Statistics of TREC Disk 5

Data Description	Size of Dataset
Foreign broadcast information service	Approx. 130,000 documents Approx. 470 MB
Los Angeles Times (from 1989 to 1990)	Approx. 130,000 documents Approx. 475 MB

Table 3. Statistics of TREC Disk 4

Data Description	**Size of Dataset**
Congressional Record of 103rd Congress	Approx. 30,000 documents Approx. 235 MB
Federal Register (1994)	Approx. 55,000 documents Approx. 395 MB

Financial Times (1992-1994)	Approx. 210,000 documents Approx. 565 MB

Table 4. Statistics of TREC Disk 2

Data Description	Size of Dataset
Wall Street Journal (1986-1989)	Approx. 100,000 documents Approx. 255 MB
AP Newswire (1989)	Approx. 85,000 documents Approx. 248 MB
Information from Computer Select disks (Ziff-Davis Publishing)	Approx. 75,000 documents Approx. 188 MB
Federal Register (1988)	Approx. 26,000 documents Approx. 211 MB

4.2 Cosine Similarity Model

There are a number of different ways to compute the similarity between documents such as cosine and correlation-based similarity. In our comparison, we use the cosine similarity-based model which is a proven IR technique in the vector space model. In the case of cosine similarity, two documents are thought of as two vectors in the *m* dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally, in the $m \times n$ ratings matrix in Fig. 2, similarity between items *i* and *j*, denoted by *sim*(*i*,*j*) is given by

$$sim\ (i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2} \quad (2)$$

where "." denotes the dot-product of the two vectors.

4.3 SLIPPER

We chose SLIPPER to compare the performance of SemanQE in generating queries. SLIPPER is an efficient rule-learning system, which is based on confidence-ruled boosting, a variant of AdaBoost [3]. 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 SLIPPER are then translated into conjunctive queries in the search engine syntax. For instance, the above rule is translated into a query "protein AND interacts."

4.4 Okapi BM25

The Okapi BM25 probabilistic model was developed by Robertson [11] and has been widely adopted in many experimental IR systems. Okapi BM25 is based on the following simple heuristics:

1) The more occurrences of a query term in a document, the more likely it is that the document is relevant.
2) A long document containing the same number of occurrences of a query term as a short one is less likely to be relevant.

The Okapi BM25 weighting function is a very well known mathematical formulation of these heuristics. The algorithms used in the experiments are denoted as follows:
BM25: The standard Okapi BM25 formula is used as the baseline:

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

(4.1)

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.

4.5 Experimental Results

We conducted a set of experiments to measure the performance of the four techniques: 1) Cosine similarity, 2) SLIPPER, 3) BM25, 4) SemanQE-Base, and 5) SemanQE-Ontologies. Because we are interested in whether ontologies have positive impact on the retrieval performance, we evaluate SemanQE in two different ways: 1) SemanQE with ontologies and 2) SemanQE without ontologies. Fig. 3 shows the results of the performance among these four techniques. The y-axis is F-measure. The F-measure combines precision and recall to provide a single number measurement for information extraction systems (3).

$$F_b = \frac{(b^2 + 1) PR}{b^2 P + R} \quad (3)$$

where P is precision, R is recall, b=0 means F = precision, b=∞ means F = recall, b=1 means recall and precision are equally weighted, b=0.5 means recall is half as important as precision. b=2.0 means recall is twice as important as precision. Because $0 \le P, R \le 1$, a larger value in the denominator means a smaller value overall.

Table 5 shows the overall performance of the four algorithms executing the query set 1-5 on TREC 5 data. The results indicate the improvements in precision at top twenty ranks (P@20) of each algorithm compared to its preceding algorithm. Among the algorithms, SemanQE with Ontologies shows the best improvement among the algorithms by achieving from 13.41% to 16.93% in terms of P@20.

Table 5. Results for TREC 5 with Four Query Expansion Algorithms by P@20

Algorithm	**TREC 5**
	P@20
SLIPPER	27.17
Cosine similarity	31.52
Okapi BM25	32.94
SemanQE+base	33.68
SemanQE+Ontologies	33.98

As shown in Fig. 3, SemanQE-Ontologies outperforms the other four techniques from 8.39% to 9.72% better in F-measure in all five cases with TREC 5. The second best technique is SemanQE-base. The performance of the BM25 technique is almost equivalent to the SemanQE-base one, which is followed by cosine similarity. SLIPPER turns out to be ranked fifth.

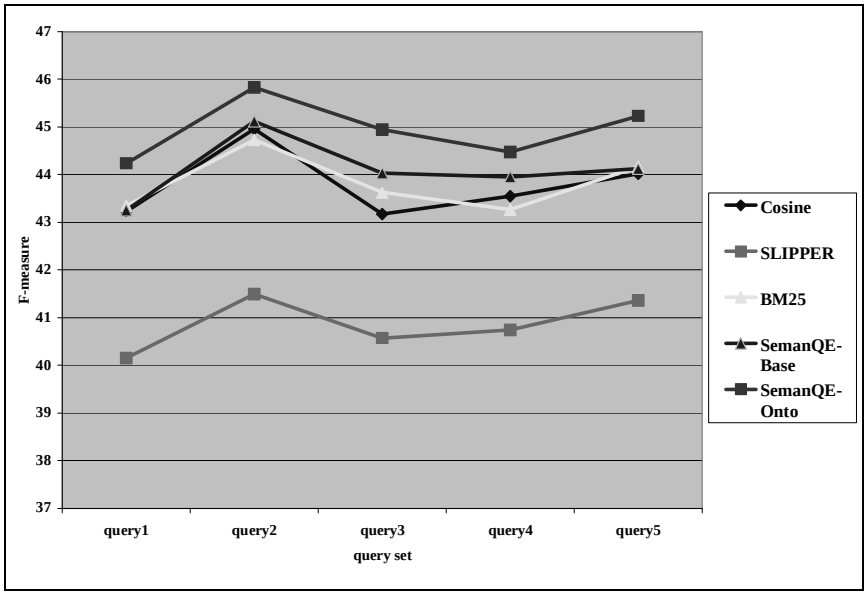

Fig. 3. Performance Comparisons among the Four Techniques on TREC-5

Table 5 indicates the overall performance of the four algorithms executing the query set 10-15 on TREC 6 data. As shown in Table 5, we observe the improvements in precision at top twenty ranked response (P@20) of each algorithm compared to its preceding algorithm. SemanQE with Ontologies achieves the best performance and outperforms by leading others from 24.09% to 32.39% better in terms of P@20.

Table 6. Results for TREC 6 with Four Query Expansion Algorithms by P@20

Algorithm	**TREC 6**
	P@20
SLIPPER	25.18
Cosine similarity	31.56
Okapi BM25	32.67
SemanQE+base	33.26
SemanQE+Ontologies	35.09

As shown in Fig. 4, the SemanQE-Ontologies method outperforms the other four techniques from 8.55% to 12.82% better in F-measure in all five cases. The second best technique is SemanQE-base. The performance of the BM25 technique is almost equivalent to the SemanQE-base one, which is followed by cosine similarity. SLIPPER turns out to be ranked fifth.

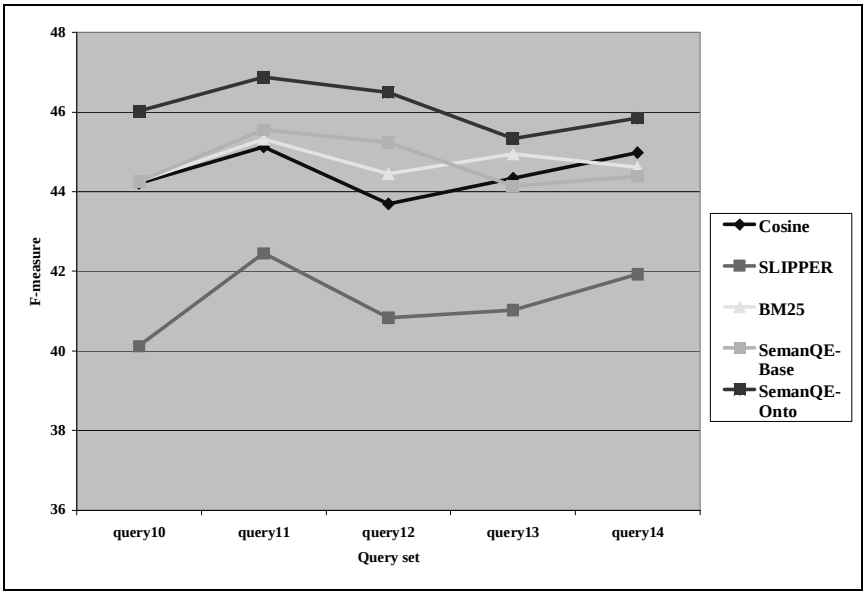

Fig. 4. Performance Comparisons among the Four Techniques on TREC-6

As with TREC 5 and 6, the experimental results indicate that the improvements in precision at top twenty ranks (P@20) of each algorithm with TREC-7 compared to its preceding algorithm. Among the algorithms, SemanQE with Ontologies in P@20 shows the best improvement among the algorithms by achieving from 17.85% to 21.51% improvement in terms of P@20.

Table 7. Results for TREC 7 with Four Query Expansion Algorithms by P@20

Algorithm	**TREC 5**
	P@20
SLIPPER	33.13
Cosine similarity	37.31
Okapi BM25	38.78
SemanQE+base	39.10
SemanQE+Ontologies	41.11

As shown in Fig. 5, the SemanQE-Ontologies method is better than the other four techniques in terms of F-measure from 10.37%% to 14.22% in all five cases. The performance of SemanQE-base and the BM25 technique are almost equivalent to each other. Cosine similarity is ranked fourth and SLIPPER is last.

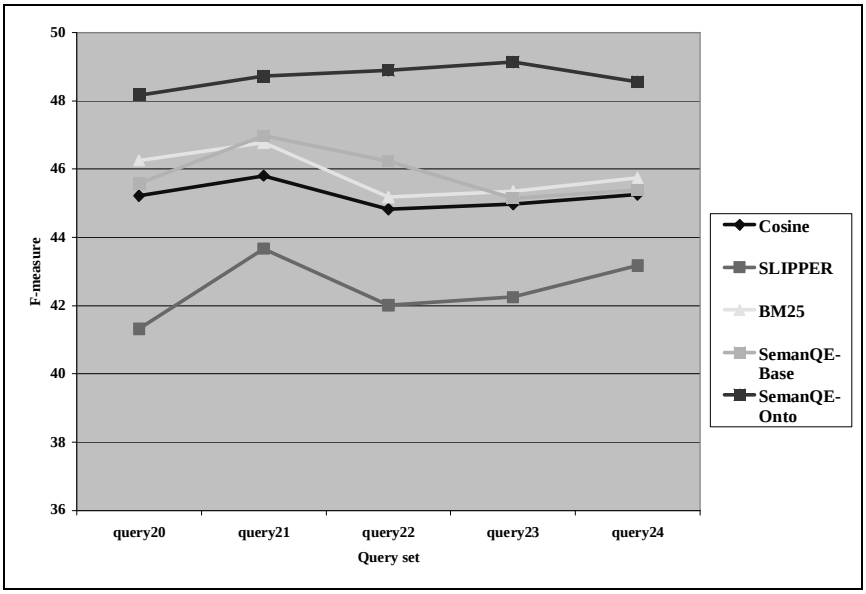

Fig. 5. Performance Comparisons among the Four Techniques on TREC-7

Overall, the results of the experiments show that SemanQE combined with ontologies achieve the best performance in both F-measure and P@20.

5 Conclusion

We proposed a novel effective query technique for information extraction, called SemanQE. SemanQE is a hybrid QE technique that applies semantic association rules to the information retrieval problem. Our approach automatically discovers the characteristics of documents that are useful for extraction of a target entity. Using these seed instances, our system retrieves a sample of documents from the database. Then we apply machine learning and information retrieval techniques to learn queries that will tend to match additional useful documents.

Our technique is different from other query expansion techniques in the following aspects. First, it proposes a hybrid query expansion algorithm combining association rules with ontologies and NLP techniques. Second, our technique utilizes semantics as well as linguistic properties of unstructured text corpus. Third, the similarity-based word sense disambiguation technique that we proposed is able to find the target sense and add semantically related ontologies' entries to queries. Fourth, our technique

makes use of contextual properties of important terms discovered by association rules.

We conducted a series of experiments to examine whether our technique improves the retrieval performance with TREC collections. We compared our technique, SemanQE+Ontologies with cosine similarity, SLIPPER, Okapi BM25, and SemanQE without Ontologies. The results show that SemanQE+Ontologies outperforms the other four techniques from 8.39% to 14.22% better in terms of F-measure in all five cases. In addition, in terms of P@20, the SemanQE+Ontologies method is significantly better than other technique from 13.41% to 32.39%.

As future studies, we will apply SemanQE to extract entity relations such as protein-protein interactions. We are interested in how SemanQE performs in discovering novel connections among the disjoint literatures where indirect connections exist among the segmented literature.

Acknowledgement

This work is supported in part by NSF Career grant IIS 0448023, NSF CCF 0514679, PA Dept of Health Tobacco Settlement Formula Grant (No. 240205 and No. 240196), and PA Dept of Health Grant (No. 239667).

References

1. Agrawal R., and Shafer J.C. Parallel mining of association rules, *IEEE Transactions on Knowledge and Data Engineering*, 8 (6): 962-969 1996.
2. 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.
3. French, J.C., Powell, A.L., Gey, F. and Perelman, N. (2001). Exploiting a Controlled Vocabulary to Improve Collection Selection and Retrieval Effectiveness. *10th International Conference on Information and Knowledge Management.*
4. Gonzalo, J., Verdejo, F., Chugur, I., and Cigarran, J., Indexing with WordNet synsets can improve text retrieval. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing systems*, Montreal 1998.
5. Lam-Adesina A.M., and Jones, G.J.F.(2001), Applying Summarization Techniques for Term Selection in Relevance Feedback, *ACM SIGIR Conference on Research and Development in Information Retrieval*: 1-9.
6. 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.
7. Latiri, C.C., Yahia, S.B., Chevallet, J.P., and Jaoua, (2003), A. *Query expansion using fuzzy association rules between terms*, in JIM'2003, France.
8. Mihalcea, R., and Moldovan, D. (2000), Semantic Indexing Using WordNet Senses. ACL *Workshop on IR & NLP*.
9. 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.
10. Ogilvie, P. and Callan, J. (2002), Experiments Using the Lemur Toolkit. In: *Proceedings of the Tenth Text Retrieval Conference*, (TREC-10), 103-108.

11. Robertson S E. On term selection for query expansion. *Journal of Documentation*, 46, 359-364, 1990.
12. 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.
13. Sanderson, M. (1994) Word sense disambiguation and information retrieval. In *Proceedings, ACM SIGIR*, pages 142-151.
14. Voorhees, E.M. (1998). Using WordNet for Text Retrieval. In *WordNet, an Electronic Lexical Database*, C. Fellbaum (ed.), MIT Press, 285-303.