Mining Dependency Relations for Query Expansion in Passage Retrieval

Renxu Sun Chai-Huat Ong Tat-Seng Chua
Department of Computer Science
School of Computing
National University of Singapore
{sunrenxu, ongchaih, chuats}@comp.nus.edu.sg

ABSTRACT

Classical query expansion techniques such as the local context analysis (LCA) make use of term co-occurrence statistics to incorporate additional contextual terms for enhancing passage retrieval. However, relevant contextual terms do not always co-occur frequently with the query terms and vice versa. Hence the use of such methods often brings in noise, which leads to reduced precision. Previous studies have demonstrated the importance of relationship analysis for natural language queries in passage retrieval. However, they found that without query expansion, the performance is not satisfactory for short queries. In this paper, we present two novel query expansion techniques that make use of dependency relation analysis to extract contextual terms and relations from external corpuses. The techniques are used to enhance the performance of density based and relation based passage retrieval frameworks respectively. We compare the performance of the resulting systems with LCA in a density based passage retrieval system (DBS) and a relation based system without any query expansion (RBS) using the factoid questions from the TREC-12 QA task. The results show that in terms of MRR scores, our relation based term expansion method with DBS outperforms the LCA by 9.81%, while our relation expansion method outperforms RBS by 17.49%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - Retrieval Models; I.2.7 [**Artificial Intelligence**]: Natural Language Processing; I.7.1 [**Document and Text Processing**]

General Terms

Algorithms, Measurement, Experimentation

Keywords

Query Expansion, Dependency Parsing, Passage Retrieval

1. INTRODUCTION

Query expansion is a widely researched topic in the field of information retrieval [1, 2, 10, 20]. It is a method for improving the effectiveness of information retrieval through the reformulation of queries by providing additional contextual information to the original queries.

Traditional passage retrieval algorithms perform a density based weighting of query terms [4, 5, 9, 16] that prefer passages containing query terms that are close together. In these density based frameworks, LCA (Local Content Analysis) [20] is a common query expansion technique based on term co-occurrence statistics. However, LCA is unable to differentiate between noisy and good quality expansion terms because it utilizes only statistical information instead of semantic information. Katz and Lin [11] pointed out the importance of relationship analysis. Firstly, relevant terms for query expansion do not necessarily always co-occur very frequently with the original query terms. Secondly, it is common to have unrelated words co-occurring with the original query terms very frequently. In order to tackle the problems of LCA for query expansion, the use of additional knowledge or linguistic cue is necessary.

Cui et al. [8] explored the use of a fuzzy dependency relation matching method to perform passage retrieval by examining the grammatical dependency relations between query terms and key terms within passages to improve passage retrieval. They found a significant increase in performance of up to 77.83% in MRR as compared to the density based passage retrieval systems. This work [8] points towards the importance of performing syntactical relational analysis to help identify good matching terms from the rest. However, they found that longer queries benefit more from the utilization of relation matching than short queries (of less than three terms) in passage retrieval. This is because short queries contain fewer contextual terms and are usually imprecise. As a result, passage retrieval loses precision on such questions. Thus, there is a need to perform query expansion that works within the framework of fuzzy relation matching to improve the performance, especially for short queries.

In addition, it was found in [1] that if we were to perform expansion on the original corpus only without utilizing additional external resources, the result obtained will be greatly dependent upon the quality of the initial retrieval. A good initial retrieval will result in an improvement in query expansion performance but a poor initial retrieval will only make it worse. Thus the use of external resources might be necessary for robust query expansion.

By taking the above issues into consideration, this paper explores the use of dependency relation analysis based on external parallel information resources to develop a robust query expansion system for passage retrieval. It performs dependency relation analysis on the passages retrieved from external corpus to identify both high quality terms and relations. We observe that the extraction of additional relations is particularly useful in supplementing the short queries in a fuzzy relation matching framework. Our results

SIGIR'06, August 6–10, 2006, Seattle, Washington, USA.

on the factoid questions[1] of TREC-12 QA task demonstrate that our approach is effective.

The main contribution of this paper is in employing a relation based model to performing: (a) contextual term selection to enhance density based passage retrieval, and (b) relation extraction to enhance the fuzzy dependency relation matching approach. Also, in order to make the expansion process more robust, it extracts relations and terms from external corpus instead of relying only on the results obtained from a local corpus.

This paper is organized as follows. In the next Section, we review related work on various query expansion techniques. Section 3 provides the details of our relation based query expansion technique. Section 4 presents our experimental results and analysis. Section 5 concludes the paper with directions for future work.

2. RELATED WORK

Statistical based query expansion is one of the earliest and classical methods of introducing additional context into a question [20]. It does not involve any form of language analysis and there are three main types, namely, local analysis [4], local context analysis [19] and global document analysis [5, 10].

In local analysis [4], the most frequent non-stop words among the top ranked passages are counted and added to the original query. This method is highly dependent on the quality of the passages retrieved in the initial retrieval. In cases where the top ranked passages retrieved have little relevance to the question, this method will not work well and it may even introduce irrelevant terms into the question and degrade the performance.

Instead of simply counting the most frequent words, local context analysis [20] counts terms in the top ranked passages that co-occur most frequently with the query terms. This method may appear to be better than local analysis as it introduces a further constraint that the expanded terms must co-occur frequently with the query terms. However, not all relevant terms for query expansion co-occur frequently with the original query terms, and that unrelated terms may co-occur very frequently with the query terms as well.

Global document analysis [5, 10] counts the most frequent terms appearing in the top ranked documents and adds them to the query. However, this approach can be expensive in terms of computation time as term searching at the document level can be inefficient. Another more serious disadvantage is that there could be more noise among the terms introduced because the analysis is done at the document level. Within a document, a word may occur very frequently in the first paragraph, while the query term may occur only in the last paragraph. Hence, there may be a lack of proper relationship between a query term and the expanded term, which may turn out to be noise instead.

A common problem with these query expansion methods is that the relationships between the original query terms and the expanded query terms are not considered. Recent studies by Katz and Lin [11] have shown that these relationships between terms are very crucial to the performance of a passage retrieval system. Therefore a good query expansion technique should make use of the relation information to provide additional contextual information to the original query. Along this direction, Cui *et al.* [8] proposed a framework for passage retrieval using dependency relation analysis. They first extracted the dependency relation paths from both the query and answer candidate sentences and ranked the answer candidate sentences according to the similarity between their relation paths with that of the query's. Similar framework has been adopted by Wu *et al.* [19], in which they tried to extract surface relation patterns from both the query and the answer candidate sentences to perform relation based matching. The main limitation with these approaches is that as they use only the dependency relations extracted from the original queries, they are ineffective for short queries that have very few query terms and relation paths. Moreover, their techniques cannot be applied to non-natural language queries as the dependency relations cannot be easily extracted from such queries.

To tackle the problems of term-based query expansion and relation-based approaches in passage retrieval as described above, we propose two methods to perform query expansion based on dependency relation analysis using external resources. The two methods are based on the extension of the technique presented in [8] to perform term expansion and relation path expansion. Term expansion is used to find expanded terms that are closely related to the original query terms, while relation path expansion aims to extract additional relations between query and expanded terms. In order to effectively apply relation-based methods to short or ungrammatical queries, we use the external resources such as the Web to extract additional terms and relations for query expansion.

3. QUERY EXPANSION BASED ON DEPENDENCY RELATION

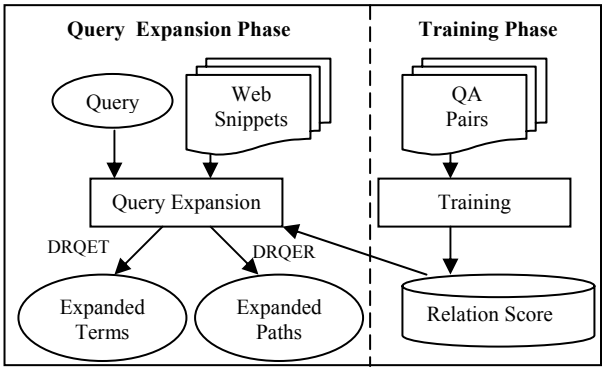

Figure 1. Framework of Relation Based Query Expansion

In this Section, we will discuss in detail how we perform query expansion using dependency relations. We first present the extraction of relation paths from parse trees in web snippets. We then describe in detail the two query expansion methods, namely: (a) dependency relation based term expansion (DRQET), which is to be employed in a density based passage retrieval system [6,9], and (b) dependency relation based path expansion (DRQER), which is to be employed in a relation based passage retrieval system [8]. Finally, we will present details on how we train our relation language model for query expansion.

In our framework for query expansion, we adopt a variation of local context method by applying language modeling techniques on relations to select the expanded terms and relation paths. Figure 1 illustrates the general framework for relation based query expansion. The framework comprises two major phases: the

[1] Factoid questions, such as "who invented the paper?", require a precise fact-oriented answers in the form of a phrase or a short passage.

training phase and the query expansion phase. During the training phase, we use the training QA pairs to derive the weights of relations between query terms and expansion terms. The relation weights are stored in the relation score table.

The query expansion phase implements two separate query expansion methods DRQET and DRQER. Both methods make use of the trained relation score table to measure the relevance of the expanded terms and relation paths from Web resources.

3.1 Dependency Relation Paths from Web Snippets

There are two sources of information corpus from which query expansion may be carried out. They are: (a) the original corpus from which information is to be found, and (b) the parallel corpus from which relevant contextual information may be mined. Original corpus is the most obvious collection and is used by most query expansion techniques based on relevance feedback [2,6,20]. Parallel corpus, which means another corpus of the same time or topic domain, is another choice and it is often adopted by news video retrieval system using ASR (Automatic Speech Recognition). With the wide spread adoption of World Wide Web, web based query expansion is adopted by most IR and Question-Answering (QA) [3] systems. There are two major reasons for using the web as a parallel corpus for IR and open domain QA: (1) the content of the web is more complete than any other existing corpus, and (2) the content of the web is dynamic and constantly being updated.

In our experiment, we use Google snippets as the basis for query expansion. We first send the queries to Google and collect the top k snippets. We adopt an approach similar to that of the local context analysis (LCA) method. In the LCA method presented in [20], top 200 passages are found to be the ideal size for query expansion. Since we are performing sentence based matching, each sentence is considered to be a passage while each snippet on average contains two complete sentences[2]. Therefore there are on average 2k passages contained in the top k snippets, and thus we set k to 100 in our experiment. There are two reasons for using snippets rather than complete html pages for passage retrieval systems. First, complete html pages can be very long and about multiple topics, it's often the case that a term at the beginning and a term at the end of a long document do not have any dependency relations. Second, it is more efficient to use snippets because we can eliminate the cost of processing the unnecessary parts of the html pages.

After we have collected the snippets, we use a sentence splitter to split the sentences within these snippets. We then parse the snippets using Minipar [14], a dependency grammar parser. We denote the question or query as Q and the set of snippets corresponding to the query as $S:=\{s_1,s_2\ldots s_m\}$.We denote the resulting set of passages (or sentences) derived from S as $P:=\{p_1,p_2\ldots p_n\}$ with the expected value of n to be around 200. We denote the set of parse trees of passages in P as $T:=\{t_1,t_2\ldots t_n\}$.

Figure 2 illustrates an example of dependency parsing. It shows the parse tree of a sample question (Q: When is Alaska purchased?) in Figure 2(a) and the parse tree of a sample answer snippet (s_i: Alaska was purchased from Russia in 1876) retrieved from Google in Figure 2(b). In a dependency tree, each node represents a word or a chunked phrase, and is attached with a link or edge representing the relation pointing from this node (the governor) to its modifier node. In this paper, we define each node in the dependency tree as a term and each edge as a dependency relation. Although dependency relations are directed links, we ignore the directions of the relations. This is because the roles of terms as governor and modifier often change in questions and answers. The label associated with the link is the type of dependency relation between two nodes. Some examples of relation labels (or relations for short) as shown in Figure 2 are obj (objective), from (relation that indicates the direction of the action) and in (relation that indicates the time information of the action). There are 42 commonly used relations defined in Minipar [14].

We further define a relation path (or simply path) between nodes n_1 and n_2 as the series of edges that traverse from n_1 to n_2. In this way, our system is able to capture long dependency relations. For simplicity, we consider a path as a vector path: <Start_Term, Rel_1,$Rel_2\ldots Rel_m$, End_Term>, where Start_Term is the starting node of the path, End_Term is the ending node and Rel_i denotes a single relation. For example, the relation path between "When" and "purchased" as shown in Figure 2(a) can be defined as Path <When, wha, head, purchased>.

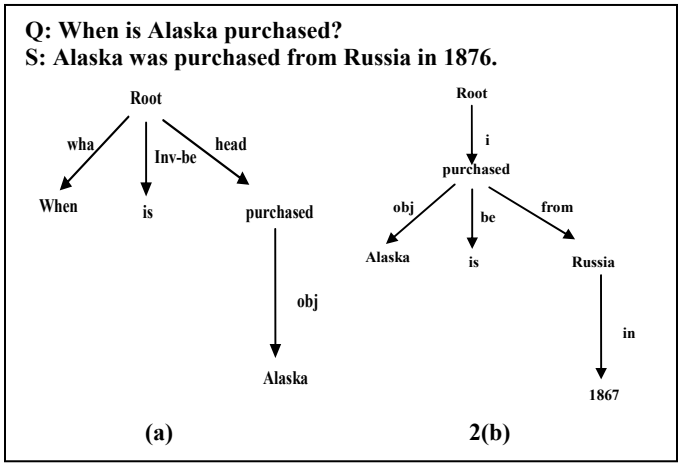

Figure 2. The parse trees of the sample question and sentence

3.2 Term Expansion for Density Based Passage Retrieval System

The goal here is to incorporate relations in the selection of additional terms for term expansion from the parse trees of the retrieved passages. Statistical co-occurrence based methods such as LCA only perform term selection based on its co-occurrence with the query terms without considering their relationships. These techniques are unable to differentiate high quality contextual terms from noise. Moreover, studies [1] showed that if we were to perform query expansion based on the original corpus only, the results is highly dependent on the quality of the initial retrieval. To make the technique more robust, we employ relation based models to select high quality terms from external information source such as the Web. A ranked list of expanded terms, with a weight associated with each term indicating its relatedness to the original query, is derived for query expansion.

In the relation based model for term expansion, the importance of a query term is determined by two factors: global importance and

[2] If the sentence is not complete, then we will locate the original complete sentence from the source html page and use it as a complete sentence.

local importance. After extracting the relation paths from web snippets, we have n dependency parsing trees in T which corresponds to n passages in P (see Section 3.1). The global importance is measured by the inverse document frequency (idf) of the expanded term, while the local importance of an expanded term is measured by its relation path linking to the query term. The overall importance of the relation path is a function of the importance of each individual relation, which is obtained through training. The assumption is that certain paths are more likely to infer a relevant expanded terms than other paths and these useful relation paths are obtained by training. The non-stop terms denoted as T_k in the snippet set S are ranked according to the formula:

$$Score(T_k, Q) = \prod_{t_i \in Q} \left(\frac{\delta + \log_{10}\left(\sum_{j=1}^{n} path\_score(T_k, t_i, j)\right) \times idf_{Tk}}{\log_{10} N} \right)^{idf_{t_i}} \quad (1)$$

where

$$path\_score(T_k, t, j) = \prod_{\substack{Tk \in p_j \wedge t \in p_j \wedge \\ \mathrm{Re}\, l_i \in path(Tk, t, j)}} score(\mathrm{Re}\, l_i)$$

$$idf_{Tk} = \max(1.0, \log_{10}(N / N_{Tk}));$$

$$idf_{t_i} = \max(1.0, \log_{10}(N / N_{t_i}));$$

T_k is the term to be ranked;

p_j is the jth passage in the passage set P;

$path(T_k,t,j)$ is the relation path in the dependency parsing tree of p_j with start node T_k and ending node t;

$path\_score\ (T_k,t,j)$ is the score of $path(T_k,t,j)$;

N is the number of passages in the snippet set S;

N_{Tk} is the number of passages in P that contains term T_k;

N_{ti} is the number of passages in P that contains term t_i;

$score(Rel_i)$ is the score of individual relation which is obtained through training, and

δ is set to 0.1 to avoid zero values.

The above formula is a variant of the term ranking formula in local context analysis [20]. In our modified model, the global importance is still modeled in the same way as the LCA method. However, we use the importance of relation path to model the local importance instead of term co-occurrence.

We treat a relation path, Path <Start_Node, Rel_1… Rel_k…Rel_m, End_Node>, as a sequence of independent relation labels. We only consider the paths where the End_Node is a query term, and the Start_Node is the term that we want to rank. Therefore by considering only the relation labels in the path, we can treat the sequence just in the same way as a word sequence and apply language model on the relation sequence, where the vocabulary of the relational language model is all the 42 possible relation labels. Thus the score of a path is proportional to the probability by which a "useful" expansion term is inferred; and this probability is calculated using the formula $\Pi score(Rel_i)$ under the assumption that each relation in the sequence is independent of eath other.

Finally, we formulate our new query Q'_t by adding the top k terms denoted as $\{T_1, T_2 \ldots T_k\}$ with k set to 10. We set the weight of the original query terms to be 1.0 and the weight of i^{th} expanded token to be (1-0.9*i/k). The new query Q'_t is a bag of terms, that can be written as <($word_1$,$weight_1$),…($word_i$,$weight_i$)>. We then issue the new query to any density based method (DBM) for passage retrieval to rank the set of passages.

3.3 Relation Path Expansion for Relation Based Passage Retrieval System

A new framework for passage retrieval based on dependency relations has been proposed in [8], in which they found that the use of dependency relations among the query terms can significantly improve the performance of passage retrieval. The framework, however, did not incorporate query expansion and it does not work well for short queries. As traditional query expansion methods only derive expanded terms without considering their relationship to the query terms, they cannot be applied directly to the fuzzy relation based framework. Here we present a technique for extracting additional relation paths from the Web, to be used on top of the relation based framework ([8]) for passage retrieval. The path expansion technique extracts additional relation paths linking the expanded terms with the original query terms. It permits the dependencies between query terms and the expanded terms to be captured.

We now describe the main stages in performing relation path expansion from external resources. First, after performing term expansion (see Section 3.2), we name the path with starting node T_k as the path associated with T_k, and we index such paths according to T_k. In Section 3.2, we have already ranked all the T_k's in S. For each T_k, we select the path associated with T_k that has the maximum $path\_score(T_k,t,j)$ to be the expanded path of T_k denoted as $path\_ex(T_k)$. The selection formula is given in Equation (2):

$$\begin{aligned} path\_ex(Tk) &= \{path(T_k, t, j) \mid path\_score(T_k, t, j) \\ &= \max_{\substack{t \in Q \\ 1 \le j \le n}} \{path\_score(T_k, t, j\}\} \end{aligned} \quad (2)$$

Second, we formulate the expanded query Q'_r as comprising the relation paths derived from the original query Q, if any, and those extracted from the external resources. We simply append the top k paths with weights (1-0.9*i/k) to the set of paths derived from the original query.

Third, we use Q'_r in a relation based method (RBM) as presented in [8] for passage retrieval. Essentially, we use the RBM to perform passage re-ranking based on the initial set of passages obtained by the density based method (DBM). For each answer candidate passage, S, we employ MiniPar to generate its dependency relation parse tree T_s. We then compute the similarity between T_s and Q'_r by first finding all possible relation path pairs from T_s and Q'_r that have the same starting and ending nodes. We then treat the matching score of a relation path from the candidate sentence as the probability of translating it to its corresponding path in the question. We denote the paired paths from the parsed query Q'_r and sentence T_s respectively as P_Q and P_S, whose lengths are represented as m and n. The translation probability $Prob(P_S|P_Q)$ is the sum over all possible alignments:

$$\Pr ob(P_S \mid P_Q) = \frac{\varepsilon}{m^n} \sum_{\alpha_1=1}^{m} \cdots \sum_{\alpha_n=1}^{m} \prod_{i=1}^{n} P_t(\mathrm{Re}\, l_i^{(S)} \mid \mathrm{Re}\, l_{\alpha_i}^{(Q)}) \quad (3)$$

where $\mathrm{Re}\, l_i^{(S)}$ stands for the i^{th} relation in path P_S and $\mathrm{Re}\, l_{\alpha_i}^{(Q)}$ is the corresponding relation in path P_Q. The alignments of relations are given by the values of α_i which indicates the corresponding relation in the question given relation $\mathrm{Re}\, l_i^{(S)}$. ε stands for a small constant. $P_t(\mathrm{Re}\, l_i^{(S)} \mid \mathrm{Re}\, l_j^{(Q)})$ denotes the relation translation

probability, i.e., relation mapping scores, which are given by a translation model learned during training, as described in [8].

3.4 Model Training

As explained in the previous Section, the relevance of the expanded term T_k is inferred by its relation paths linking it to the query terms. To avoid the data sparseness problem in training, we assume that each relation appears independently of the other relations in the same path. Hence, we have:

$$path\_score(T_k, t) = \prod_{\mathrm{Re}\, l_i \in path(Tk,t)} score(\mathrm{Re}\, l_i) \quad (4)$$

Therefore for each type of relation in the dependency parsing tree, we need to estimate *score(Rel$_i$)* from the training corpus.

To perform training, we use the TREC 8 and TREC 9 QA question-answer pairs as the training set. We denote each QA pair as (Q_i, A_i). We retrieve the top 100 snippets from Google for each question, perform sentence splitting and dependency parsing, and select the "relevant" paths from the set of parsing trees of the snippets. A path p in the snippets corresponding to Q_i (denoted as <Start_Node, Rel_1… Rel_k…Rel_m, End_Node>) is relevant if $Start\_Node \in A_i$ and $End\_Node \in Q_i$. In other words, the relevant paths are those inferring a useful term to the question. After collecting all the relevant paths, we employ a unigram language model [15] to train the weight of individual relations. Relation labels are treated as vocabularies in a language model. Therefore, the score of individual relation should be proportional to the probability of such a relation appearing in the training data set as shown in Equation (5). We use the smoothed probability to avoid zero values and take the log of frequency count to reduce the variance of the score. The eventual formula used to calculate the final score is given in Equation (6).

$$P(\mathrm{Re}l_i) = C_{\mathrm{Re}l_i \in relevant\_path} + 1/((\sum_{1 \le i \le N} C_{\mathrm{Re}l_i \in relevant\_path}) + N) \quad (5)$$

$$score(\mathrm{Re}\ l_i) = \log(C_{\mathrm{Re}l_i \in relevant\_path} + 1) / \log((\sum_{1 \le i \le N} C_{\mathrm{Re}l_i \in relevant\_path}) + N) \quad (6)$$

where $C_{\mathrm{Re}l_i \in relevant\_path}$ is the number of $\mathrm{Re}l_i$ in relevant paths, and N is the total number of relation types.

4. EVALUATIONS

In this Section, we present the empirical results to evaluate our query expansion techniques for passage retrieval. Our evaluations aim to verify three hypotheses as follows:

(1) It is effective to incorporate dependency relation based query expansion technique to select high quality terms in a density based passage ranking framework.

(2) The use of dependency relation based query expansion technique to extract relation paths further improves the precision of passage ranking when integrated with fuzzy relation matching technique.

(3) As short queries with fewer key terms are more likely to have word mismatch problems when performing passage retrieval, the short queries will benefit more from dependency based query expansion.

4.1 Experiment Setup

We accumulate 10,255 factoid question-answer pairs from the TREC-8 and TREC-9 QA tasks. For each question, we collect the top 100 snippets from Google, from which we extract 8,892 relevant paths used for training of individual relation weights using the unigram language model.

We use the factoid questions from the TREC-12 QA task [17] for testing data and the AQUAINT corpus[3] to search for the answers. After excluding 30 questions that have NIL answers and 59 questions that do not have any ground truth passages, we obtain 324 factoid questions from TREC-12 task. Similar to the configuration used by Tellex *et al.* [16] and Cui *et al.* [8], we use the top 200 documents for each question based on the retrieved document list provided by TREC as the basis to construct the relevant document set for the questions. If these 200 documents do not contain the correct answer, we add supporting documents that have the answer into the document set. We conduct different combination of query expansion and passage retrieval algorithms on the document set to return the top 20 ranked passages. In order to make the results comparable to [8] we use exactly the same testing data set and system configuration as in [8].

We implement a state-of-the-art density based system (DBS) based on [9] and a corresponding EM-based fuzzy relation matching system (RBS) based on ([8, 19]) for passage retrieval. We also implement three query expansion algorithms: (a) LCA [20]; (b) a dependency relation based technique for term expansion (DRQET); (c) a dependency relation based technique for relation path expansion (DRQER). It is noted that the density based system RBS can only employ the term expansion technique (DRQET); while the relation based system (RBS) may employ only the path expansion technique (DRQER). Future work will examine how the two query expansion techniques may be integrated into the same framework.

We arrive at the following five comparison systems:

1. DBS [9]: A density based passage retrieval system and is used as a baseline.

2. DBS+LCA: DBS system integrated with LCA (local context analysis method) for query expansion [20] based on top 100 Google snippets.

3. DBS+DRQET: DBS system integrated with dependency relation based query expansion (DRQET for term ranking only) based on top 100 Google snippets.

4. RBS[8]: A passage ranking algorithm based on fuzzy relation matching.

5. RBS+DRQER: RBS integrated with dependency relation based path expansion (DRQER).

[3] This corpus consists of over 1 million of newswire text articles in English for the period form 1996-2000. The articles are drawn from three sources: the Xinhua News Service, the New York Times News Service, and the Associated Press News Service. It is used in official benchmark evaluations conducted by National Institute of Standards and Technology (NIST)

Table 1. Overall performance comparison of MRR, percentage of incorrectly answered questions (% Incorrect) and precision at top one passage of the five systems. All improvements are statistically significant ($p<0.01$).

Passage retrieval systems	DBS	DBS+LCA	DBS+DRQET	RBS	RBS+DRQER
MRR	0.2677	0.3293	0.3616	0.4761	0.5541
% MRR improvement over					
DBS	N/A	+23.01	+35.08	+77.83	+106.99
DBS+LCA	N/A	N/A	+9.81	+44.58	+68.27
RBS	N/A	N/A	N/A	N/A	+17.49
% Incorrect	33.02%	28.40%	27.16%	24.07%	21.60%
Precision at top one passage	0.1759	0.2315	0.2840	0.3889	0.4228

Table 2. Overall performance comparison of MRR, percentage of incorrectly answered questions (% Incorrect) and precision at top one passage of the five systems tested on dataset D2 containing queries with less than or equal to 3 non-trivial terms.

Passage retrieval systems	DBS	DBS+LCA	DBS+DRQET	RBS	RBS+DRQER
MRR	0.1812	0.2413	0.2816	0.2570	0.3314
% MRR improvement over					
DBS	N/A	+33.17	+55.40	+41.83	+82.89
DBS+LCA	N/A	N/A	+16.70	+6.50	+37.34
RBS	N/A	N/A	N/A	N/A	+28.94
% Incorrect	48.31%	41.85%	33.70%	38.48%	29.78%
Precision at top one passage	0.1096	0.1657	0.2022	0.1741	0.2612

We employ three performance metrics: mean reciprocal rank (MRR), percentage of questions that have no correct answers and precision at the top one passage. The first two metrics are calculated based on the top 20 returned passages by each system.

4.2 Evaluation of Two Relation Based Query Expansion Techniques

In the first experiment, we evaluate the overall performance of our query expansion techniques when integrated into different passage retrieval systems based on the 324 factoid questions selected from TREC-12.

We apply LCA, DRQET on DBS and DRQER on RBS and list the evaluation results in Table 1. From the Table, we draw the following observations:

(1) By comparing the MRR scores between DBS and DBS+LCA, we observe a 23% improvement with the use of a term co-occurrence based query expansion technique. This confirms that query expansion can indeed improve the performance of passage retrieval.

(2) We observe an additional 9.81% improvement in MRR when comparing the performance DBS+LCA and DBS+DRQET. This shows that dependency relation based term expansion (DRQET) significantly outperforms the local context method under a density based passage retrieval framework.

(3) Under the relation based framework for passage retrieval, dependency relation based path expansion can further bring about a 17.49% improvement in MRR over fuzzy matching (RBS) of relation matching without any query expansion. This improvement is significant as RBS is already attaining a high performance of 0.4761 in MRR score. The use of relation path query expansion (DRQER) under RBS can further improve the MRR score to over 0.554, which is significantly better than the best reported results in [8] for RBS without query expansion.

We also observe the same trends of improvement on "precision at top one passage" and reduction in "percentage of incorrect passages" when comparing the results of passage retrieval with and without relation based query expansion.

4.3 Evaluation of Variations in Query Length

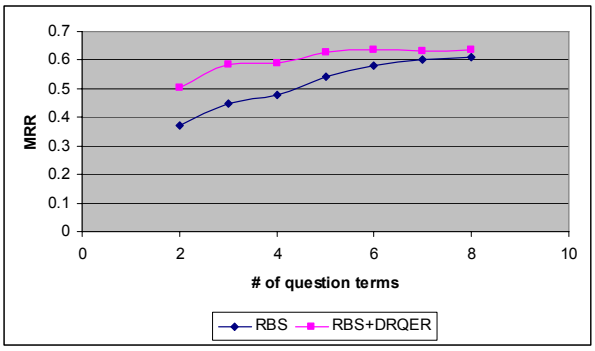

Figure 2. MRR before and after query expansion vs number of non-trivial question terms.

Earlier research on local context analysis framework [20] showed that query expansion is most useful for short queries; while work on fuzzy relation matching without query expansion [8] concludes that longer queries are more likely to benefit from relation matching. We want to verify whether such conclusions still hold in relation based query expansion. Figure 2 plots the performance in terms of MRR for RBS and RBS+DRQER for queries with the number of non-trivial terms varying from two to eight. Figure 2 shows that query expansion can bring more than 30% of improvement for queries with less than three terms. However, as the number of query terms increases, the rates of improvement brought about by query expansion become significantly less.

The reason why query expansion is more useful for short queries is that query expansion is employed to overcome the problems of the lack of context and word mismatch in passage retrieval.

The severity of the problem tends to decrease as queries get longer, as there is more chance of some important words co-occurring in the query and relevant documents. So query expansion may be less effective in cases of longer queries with over six non-trivial terms.

4.4 Evaluation of Short Queries

In this Section we aim to conduct a more comprehensive test on short queries comprising three or less non-trivial question terms to simulate web queries. For this, we set up a query set D2, which comprises 356 short queries obtained from the factoid questions in TREC-11 [18] and TREC-12 QA task [16] after filtering out questions with more than three non-trivial terms. We use the same five comparison systems as described in Section 4.2 and present the result in Table 2. From Table 2, we can see that the improvement is even more significant on short queries. We observe a 16.7% of improvement in MRR by comparing DBS+DRQET with DBS+LCA. This indicates that even without considering language constructs in the question, relation based query expansion can still perform better than co-occurrence based query expansion. This result may have implications to Web queries which tend to be short and are not in natural language format. Therefore, future query expansion algorithms should incorporate some form of relation analysis rather than simply counting the co-occurrence between expansion terms and query terms.

Interestingly, for short queries we find that relation matching without query expansion (RBS) performs worse than a density based passage ranking with dependency based query expansion (DBS+DRQET). This shows that query expansion is crucial for short queries as it is hard to extract word dependency information from the original query for RBS. With the use of relation path expansion, RBS+DRQER again performs the best among all the five systems for both short queries and long queries. This is because RBS+DRQER expands the query terms first and then resolve the dependency relations between query terms and the expansion terms. This result indicates that incorporating dependency relation analysis into both query expansion and passage ranking will boost the performance of passage retrieval.

Analysis of results in terms "% Incorrect" and "Precision at top one passage" shows similar trends as the MRR scores.

4.5 Sample Case Analysis

In this Section, we will use two examples to illustrate the advantage of incorporating dependency relation analysis into query expansion. In the first example, we will illustrate that relation based term expansion is more effective than co-occurrence based term expansion. In the second example, we will show that relation path expansion is often necessary than merely performing term expansion.

Example 1: Question: "What did George Washington call his house?" Figure 3 shows the top 10 expanded terms by LCA (3(a)) and DRQET (3(b)). We observe from Figure 3 that the expanded terms provided by LCA are related to either the concepts "George Washington" or "house". However, very few terms are related to both concepts in the given query. The main reason is that LCA only counts the statistical co-occurrence between query terms and expanded terms independently without considering their specific dependency relationship. If we were to perform dependency relation analysis on the expanded terms listed in Figure 3(a), we may find that the dependencies between some of the expanded terms and query terms may not be that strong. For example, the word "white" only has close dependency with " house" but it does not have close dependency with "George Washington" in the web snippets. On the other hand, when we examine the expanded terms given by DRQET, we find that although some terms such as "Mount" and "Vernon" only co-occur five times with query terms among top the 100 snippets, they are still ranked at top positions due to their strong and close relationship with all the query terms. For instance, in one of the snippets says "George's Washington's house, Mount Vernon". The dependency relation between "house" and "Mount Vernon" is "appo" (appositive), which is a very strong relationship according to the trained relation score indicating that this relation is likely to infer relevant words.

(a) Top 10 Expanded Terms (LCA)	(b) Top 10 Expanded Terms (DRQET)
President	President
life	Mount
Martha	Vernon
white	Martha
command	1732
father	United States
cherry	American
1732	command
United States	revolutionary
war	war

Figure 3. Illustration of expanded terms provided by LCA and DRQET of the sample question

In the second example, we consider the question "What country made the Statue of Liberty?" Since "Statue of Liberty" is a very popular term, it will dominate the whole query. As a result, both LCA and DRQET (term expansion) will expand terms such as "France", "New York", "Paris" and "USA". If we simply use these terms in passage retrieval, we will find that the following answer candidates are likely to obtain the same score: (a) Statue of Liberty is built in France, and (2) Statue of Liberty is now in New York. This is because both answer candidates have almost the same key word density. However, if we further consider the relation path between "Statue of Liberty" and "France" (or "New York") in the two answer candidates, we will observe that the first sentence is more relevant than the second one as the relation path <Statue of Liberty, obj, in, France> is matched with the path <Statue of Liberty, obj, in, LOC_COUNTRY>. Hence, by expanding and matching the relation path between terms can further improve the accuracy of passage retrieval.

5. CONCLUSION

In this paper, we presented two dependency relation based query expansion techniques for passage retrieval. The first technique makes use of relation analysis to extract high quality contextual terms for use in a density based passage retrieval framework. The second technique extracts relation paths for query expansion in a relation based passage retrieval framework [8].

Evaluation results showed that our first technique used in conjunction with the density based frameworks produces a

significant improvement of 9.81% in retrieval performance as compared to LCA. Also, our second technique used in conjunction with the relation based frameworks produced a 17.49% improvement over a corresponding relation based passage retrieval system without query expansion. In this paper, we also studied the relationship between query lengths and improvements by query expansion. Our experiment showed that short queries tend to benefit more from query expansion. To ascertain this, we conducted another experiment particularly for short queries. The result showed even more significant improvement: a 16.70% improvement over a density based passage retrieval algorithm with our first technique, and a 28.94% improvement over a passage retrieval system using dependency relation matching with our second technique.

In the second experiment, we also observed the drawback of fuzzy dependency matching on short queries, which have very few relations between terms. Therefore we believed that additional contextual relations should be introduced in order to achieve better recall of matching. The experimental result showed that our second technique that expands queries in relation based frameworks (RBS+DRQER) again performs the best among all the five comparison systems.

In the future, we will continue our research work on relation based models for information retrieval. In particular, we will continue our research in the following directions: (1) explore the use of different models and their combinations for relation based query expansion for passage retrieval; and (2) conduct detailed analysis on the performance of dependency relation based query expansion on different types of queries.

6. REFERENCES

[1] G. Amati, C. Carpineto, G. Romano, *Query Difficulty, Robustness, and Selective Application of Query Expansion*. ECIR 2004, pp. 127-137

[2] R. Attar, A. S. Fraenkel, (1977). *Local Feedback in Full-Text Retrieval Systems,* Journal of the Association for Computing Machinery, 24(3), pp. 397-417.

[3] E. Brill, J. Lin, M. Banko, Susan T. Dumais, A. Ng: *Data-Intensive Question Answering.* Proceedings of TREC-10, 2001 pp.393-400.

[4] C. Buckley, A. Singhal, M. Mitra, G. Salton, *New Retrieval Approaches Using SMART: TREC 4,* Proceedings of the TREC 4 Conference.

[5] J. Callan, W. B. Croft, J. Broglio, *TREC and TIPSTER experiments with INQUERY,* Information Processing and Management 1995, pp. 327-343.

[6] W. B. Croft, D. J. Harper, (1979). *Using probabilistic models of document retrieval without relevance information,* Journal of Documentation, 35, pp. 285-295.

[7] W. B Croft, R. Cook, D. Wilder, *Providing Government Information on The Internet: Experiences with THOMAS,* In Digital Libraries Conference DL'95, pp. 19-24.

[8] H. Cui, R. Sun, K. Li, M.-Y. Kan and T.-S. Chua. *Question Answering Passage Retrieval Using Dependency Relations,* Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil , Aug 15-19, pp. 400 - 407.

[9] H. Cui, K. Li, R. Sun, T.-S. Chua and M.-Y. Kan, *National University of Singapore at the TREC-13 Question Answering Main Task*, Proceedings of TREC-13, 2004.

[10] Y. Jing and W. B. Croft, *An Association Thesaurus for Information Retrieval,* Proceedings of RIAO 94, pp. 146-160.

[11] B. Katz and J. Lin, *Selectively Using Relations to Improve Precision in Question Answering,* Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering, April 2003

[12] G. G. Lee, J. Seo, S. Lee, H. Jung, B.-H. Cho, C. Lee, B.-K. Kwak, J. Cha, D. Kim, J. An, H. Kim, and K. Kim, *SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP*, Proceedings of TREC-10, 2001, pp. 442-451.

[13] D. Lin and P. Pantel, *Discovery of Inference Rules for Question Answering,* Natural Language Engineering, 2001, 7(4): pp. 343-360.

[14] D. Lin, *Dependency-based Evaluation of MINIPAR,* Proceedings of Workshop on the Evaluation of Parsing Systems, Granada, Spain, May, 1998.

[15] F. Song and B. Croft, *A general language model for information retrieval*, Proceedings of CIKM'99, 1999, pp. 316-321.

[16] S. Tellex, B. Katz, J. Lin, A. Fernandes and G. Marton, *Quantitative evaluation of passage retrieval algorithms for question answering*, Proceedings of SIGIR '03, 2003, Toronto, Canada, pp. 41-47.

[17] E. M. Voorhees, *Overview of the TREC 2003 Question Answering Track,* Proceedings of TREC-12, pp. 54-68.

[18] E. M. Voorhees, Overview of the TREC 2002 Question Answering Track, Proceedings of TREC-12, pp. 60-71.

[19] M. Wu, M. Duan, S. Shaikh, S. Small, T. Strzalkowski *University of Albany's ILQUA in TREC 2005*, Proceedings of TREC-14 2005 pp.77-83.

[20] J. Xu, W. B. Croft, *Query expansion using local and global document analysis,* Proceedings of the 19th annual international ACM SIGIR 1996 conference on Research and development in information retrieval , Zurich, Switzerland, pp. 4 – 11.