Context-Dependent Term Relations for Information Retrieval

Jing Bai Jian-Yun Nie Guihong Cao
DIRO, University of Montreal
CP. 6128, succ. Centre-ville, Montreal,
Quebec H3C 3J7, Canada
{baijing,nie,caogui}@iro.umontreal.ca

Abstract

Co-occurrence analysis has been used to determine related words or terms in many NLP-related applications such as query expansion in Information Retrieval (IR). However, related words are usually determined with respect to a single word, without relevant information for its application context. For example, the word "*programming*" may be considered to be strongly related to "*Java*", and applied inappropriately to expand a query on "*Java travel*". To solve this problem, we propose to add another context word in the relation to specify the appropriate context of the relation, leading to term relations of the form "(*Java*, *travel*) → *Indonesia*". The extracted relations are used for query expansion in IR. Our experiments on several TREC collections show that this new type of context-dependent relations performs much better than the traditional co-occurrence relations.

1. Introduction

A query usually is a poor expression of an information need. This is not only due to its short length (usually a few words), but also due to the inability of users to provide the best terms to describe their information need. At best, one can expect that some, but not all, relevant terms are used in the query. Query expansion thus aims to improve query expression by adding related terms to the query. However, the effect of query expansion is strongly determined by the term relations used (Peat and Willett, 1991). For example, even if "*programming*" is strongly related to "*Java*", if this relation is used to expand a query on "*Java travel*", the retrieval result will likely deteriorate because the irrelevant term "*programming*" is introduced, leading to the retrieval of irrelevant documents about "*programming*".

A number of attempts have been made to deal with the problem of selecting appropriate expansion terms. For example, Wordnet has been used in (Voorhees, 1994) to determine the expansion terms. However, the experiments did not show improvement on retrieval effectiveness. Many experiments have been carried out using associative relations extracted from term co-occurrences; but they showed variable results (Peat and Willett, 1991). In (Qiu and Frei, 1993), it is observed that one of the reasons is that one tried to determine expansion terms according to each original query term separately, which may introduce much noise. Therefore, they proposed to determine the expansion terms by summing up the relations of a candidate expansion term to each of the query terms. In so doing, a candidate expansion term is preferred if it has a strong relationship with many of the query terms. However, it is still difficult to prevent the expansion process from adding "*programming*" to a query on "*Java travel*" because of its very strong relation with "*Java*".

The approach used in (Qiu and Frei, 1993) indeed tries to correct a handicap inherent in the relations: as term relations are created between two single words such as "*Java* → *programming*", no information is available to help determine the appropriate context to apply it. The approach used in (Qiu and Frei, 1993) can simply alleviate the problem without solving it radically.

In this paper, we argue that the solution lies in the relations themselves. They have to contain more information to help determine the appropriate context to apply them. We thus propose a way to add some context information into the relations: we introduce an additional word into the condition part of the relation, such as "(*Java*, *computer*) → *programming*", which

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 551–559,
Sydney, July 2006.

means "*programming*" is related to "(*Java*, *computer*)" together. In so doing, we would be able to prevent from extracting and applying a relation such as "(*Java*, *travel*) → *programming*".

In this paper, we will test the extracted relations in query expansion for IR. We choose to implement query expansion within the language modeling (LM) framework because of its flexibility and high performance. The experiments on several TREC collections will show that our query expansion approach can bring large improvements in retrieval effectiveness.

In the following sections, we will first review some of the relevant approaches on query expansion and term relation extraction. Then we will describe our general IR models and the extraction of term relations. The experimental results will be reported and finally some conclusions will be drawn.

2. Query Expansion and Term Relations

It has been found that a key factor that determines the effect of query expansion is the selection of appropriate expansion terms (Peat and Willett, 1991). To determine expansion terms, one possible resource is thesauri constructed manually, such as Wordnet. Thesauri contain manually validated relations between terms, which can be used to suggest related terms. (Voorhees, 1994) carried out a series of experiments on selecting related terms (e.g. synonyms, hyonyms, etc.) from Wordnet. However, the experiments did not show that this can improve retrieval effectiveness. Some of the reasons are as follows: Although Wordnet contains many relations validated by human experts, the coverage is far from complete for the purposes of IR: not only linguistically motivated relations, but also association relations, are useful in IR. Another problem is the lack of information about the appropriate context to apply relations. For example, Wordnet contains two synsets for "*computer*", one for the sense of "*machine*" and another for "*human expert*". It is difficult to automatically select the correct synset to expand the word "*computer*" even if we know that the query's area is computer science.

Another often used resource is associative relations extracted from co-occurrences: two terms that co-occur frequently are thought to be associated to each other (Jing and Croft, 1994). However, co-occurrence relations are noisy: Frequently co-occurring terms are not necessarily related. On the other hand, they can also miss true relations. The most important problem is still that of ambiguity: when one term is associated with another, it may be related for one sense and not for other possible senses. It is then difficult to determine when the relation applies.

In most of the previous studies, relations extracted are restricted between one word and another. This limitation makes the relations ambiguous, and their utilization in query expansion often introduces undesired terms. We believe that the key to make a relation less ambiguous is to add some contextual information.

In an attempt to select better expansion terms, (Qiu and Frei, 1993) proposed the following approach to select expansion terms: terms are selected according to their relation to the whole query, which is calculated as the sum of their relations to each of the query terms. Therefore, a term that is related to several query terms will be favored. In a similar vein, (Bai et al. 2005) also try to determine the relationship of a word to a group of words by combining its relationships to each of the words in the group. This can indeed select better expansion terms. The consideration of other query terms produces a weak contextual effect. However, this effect is limited due to the nature of the relations extracted, in which a term depends on only one other term. Much of the noise in the sets will remain after selection.

For a query composed of several words, what we would really like to have is a set of terms that are related to all the words taken together (and not separately). By combining words in the condition part such as "(*Java, travel*)" or "(*base, bat*)", each word will serve as a context to the other in order to constrain the related terms. In these cases, we would expect that "*hotel*", "*island*" or "*Indonesia*" would co-occur much more often with "(*Java*, *travel*)" than "*programming*", and "*ball*", "*catcher*" etc. co-occur much more often with "(*base, bat*)" than "*animal*" or "*foundation*".

One naturally would suggest that compound terms can be used for this purpose. However, for many queries, it is difficult to form a legitimate compound term. Even if we can detect one occurrence of a compound, we may miss others that use its variants. For example, if "*Java travel*" is used as a query, we will likely be able to consider it as a compound term. The same compound (or its variant) would be difficult to

detect in a document talking about traveling to Java: the two words may appear at some distance or not in some specific syntactic structure as required in (Lin, 1997). This will lead to the problem of mismatching between document and query.

In fact, compound terms are not the only way to add contextual information to a word. By putting two words together (without forming a compound term), we usually obtain a more precise sense for each word. For example, from "*Java travel*", we can guess that the intended meaning is likely related to "*traveling to Java Island*". People will not interpret this combination in the sense of "*Java programming*". In the same way, people would not consider "*animal*" to be a related term to "*base, bat*". These examples show that in a combination of words, each word indeed serves to specify a context to interpret another word. It then suggests the following approach: we can adjunct some additional word(s) in the condition part of a relation, such as "(*Java*, *travel)* → *Indonesia*", which means "*Indonesia*" is related to "(*Java, travel*)" together. It is expected that one would not obtain "(*Java*, *travel)* → *programming*".

Owing to the context effect explained above, we will call the relations with multiple words in the condition part *context-dependent* relations. In order to limit the computation complexity, we will only consider adding one additional word into relations.

The proposed approach follows the same principle as (Yarowsky, 1995), which tried to determine the appropriate word sense according to one relevant context word. However, the requirement for query expansion is less than word sense disambiguation: we do not need to know the exact word sense to make expansion. We only need to determine the relevant expansion terms. Therefore, there is no need to determine manually a set of seeds before the learning process takes place.

To some extent, the proposed approach is also related to (Schütze and Pedersen, 1997), which calculate term similarity according to the words appearing in the same context, or to second-order co-occurrences. However, a key difference is that (Schütze and Pedersen, 1997) consider only separate context words, while we consider multiple context words together.

Once term relations are determined, they will be used in query expansion. The basic IR process will be implemented in a language modeling framework. This framework is chosen for its flexibility to integrate term relations. Indeed, the LM framework has proven to be capable of integrating term relations and query expansion (Bai et al., 2005; Berger and Lafferty, 1999; Zhai and Lafferty, 2001). However, none of the above studies has investigated the extraction of strong context-dependent relations from text collections.

In the next section, we will describe the general LM framework and our query expansion models. Then the extraction of term relation will be explained.

3. Context-Dependent Query Expansion in Language Models

The basic IR approach based on LM (Ponte and Croft, 1998) determines the score of relevance of a document D by its probability to generate the query Q. By assuming independence between query terms, we have:

$$P(Q \mid D) = \prod_{w_i \in Q} P(w_i \mid D) \propto \sum_{w_i \in Q} \log P(w_i \mid D)$$

where $P(w_i \mid D)$ denotes the probability of a word in the language model of the document D. As no ambiguity will arise, we will use D to mean both the language model of the document and the document itself (similarly for a query model and a query Q).

Another score function is based on KL-divergence or cross entropy between the document model and the query model:

$$score(D,Q) = \sum_{w_i \in V} P(w_i \mid Q) \log P(w_i \mid D)$$

where V is the vocabulary. Although we have both document and query models in the above formulation, usually only the document model is smoothed, while the query model uses Maximum Likelihood Estimation (MLE) $P_{ML}(w_i \mid Q)$. Then we have:

$$score(D,Q) = \sum_{w_i \in Q} P_{ML}(w_i \mid Q) \log P(w_i \mid D)$$

However, it is obvious that a distance (KL-divergence) measured between a short query of a few words and a document cannot be precise. A better expression would contain all the related terms. The construction of a better query expression is the very motivation for query expansion in traditional IR systems. It is the same in LM for IR: to create a better query expression (model) to be able to measure the distance to a

document in a more precise way. The key to creating the new model is the integration of term relations.

3.1 LM for Query Expansion

Term relations have been used in several recent language models in IR. (Berger and Lafferty, 1999) proposed a translation model that expands the document model. The same approach can also be used to expand the query model. Following (Berger and Lafferty, 1999), we arrive at the first expansion model as follows, which has also been used in (Bai et al., 2005):

Model 1: Context-independent query expansion model (CIQE)

$$P_R(w_i \mid Q) = \sum_{q_j \in V} P_R(w_i, q_j \mid Q) = \sum_{q_j \in Q} P_R(w_i \mid q_j) P_{ML}(q_j \mid Q)$$

In this model, each original query term q_j is expanded by related terms w_i. The relations between them are determined by $P_R(w_i \mid q_j)$. We will explain how this probability is defined in Section 3.2. However, we can already see here that w_i is determined solely by one of the query term q_j. So, we call this model "context-independent query expansion model" (CIQE).

The above expanded query model enables us to obtain new related expansion terms, to which we also have to add the original query. This can be obtained through the following smoothing:

$$P(w_i \mid Q) = \lambda_1 P_{ML}(w_i \mid Q) + (1-\lambda_1) \sum_{q_j \in Q} P_R(w_i \mid q_j) P_{ML}(q_j \mid Q) \quad (1)$$

where λ_1 is a smoothing parameter.

However, if the query model is expanded on all the vocabulary (V), the query evaluation will be very time consuming because the query and the document have to be compared on every word (dimension). In practice, we observe that only a small number of terms have strong relations with a given term, and the terms having weak relations usually are not truly related. So we can limit the expansion terms only to the strongly related ones. By doing this, we can also expect to filter out some noise and considerably reduce the retrieval time.

Suppose that we have selected a set E of strong expansion terms. Then we have:

$$score(D,Q) = \sum_{w_i \in V} P(w_i \mid Q) \log P(w_i \mid D) \approx \sum_{w_i \in E \cup Q} P(w_i \mid Q) \log P(w_i \mid D)$$

This query expansion method uses the same principle as (Qiu and Frei, 1993), but in a LM setting: the selected expansion terms are those that are strongly related to all the query terms (this is what the summation means). The approach used in (Bai et al., 2005) is slightly different: A context vector is first built for each word; then a context vector for a group of words (e.g. a multi-word query) is composed from the context vectors of the words of the group; finally related terms to the group of words are determined according to the similarity of their context vectors to that of the group. This last step uses second-order co-occurrences similarly to (Schütze and Pedersen, 1997). In both (Qiu and Frei, 1993) and (Bai et al., 2005), the terms related to a group of words are determined from their relations to each of the words in the group, while the latter relations are extracted separately. Irrelevant expansion terms can be retained.

As we showed earlier, in many cases, when one additional word is used with another word, the sense of each of them can usually be better determined. This additional word may be sufficient to interpret correctly many multi-word user queries. Therefore, our goal is to extract stronger context-dependent relations of the form $(q_j \; q_k) \rightarrow w_i$, or to build a probability function $P_R(w_i \mid q_j q_k)$. Once this function is determined, it can be integrated into a new language model as follows.

Model 2: Context-dependent query expansion model (CDQE)

$$P_R(w_i \mid Q) = \sum_{q_j, q_k \in V} P_R(w_i \mid q_j q_k) P(q_j q_k \mid Q) \approx \sum_{q_j, q_k \in Q} P_R(w_i \mid q_j q_k) P(q_j q_k \mid Q)$$

As $P_R(w_i \mid q_j q_k)$ is a relation with two terms as condition, we will also call it a *biterm* relation. The name "biterm" is due to (Srikanth and Srihari, 2002), which means two terms co-occurring within some distance. Similarly, $P_R(w_i \mid q_j)$ will be called *unigram* relation. The corresponding query models will be called biterm relation model and unigram relation model.

As in general LM, the biterm relation model can be smoothed with a unigram model. Then we have the following score function:

$$P_R(w_i \mid Q) = \lambda_2 P_{ML}(w_i \mid Q) + (1-\lambda_2) \sum_{q_j, q_k \in Q} P_R(w_i \mid q_j q_k) P(q_j q_k \mid Q) \quad (2)$$

where λ_2 is another smoothing parameter.

3.2 Extraction of Term Relations

The key problem now is to obtain the relations we need: $P_R(w_i \mid w_j)$ and $P_R(w_i \mid w_j w_k)$. For the first probability, as in many previous studies, we exploit term co-occurrences. $P_R(w_i \mid w_j)$ could be built as a traditional bigram model. However, this is not a good approach for IR because two related terms do not necessarily co-occur side by side. They often appear at some distance. Therefore, this model is indeed a *biterm* model (Srikanth and Srihari, 2002), i.e., we allow two terms be separated within some distance. We use the following formula to determine this probability:

$$P_R(w_i \mid w_j) = \frac{c(w_i, w_j)}{\sum_{w_l} c(w_l, w_j)}$$

where $c(w_i, w_j)$ is the frequency of co-occurrence of the biterm (w_i, w_j), i.e. two terms in the same window of fixed size across the collection. In our case, we set the window size at 10 (because this size turned out to be reasonable in our pilot experiments).

For $P(w_i \mid w_j w_k)$, we further extend the biterm to triterm, and we use the frequency of co-occurrences of three terms $c(w_i, w_j, w_k)$ within the same windows in the document collection:

$$P_R(w_i \mid w_j w_k) = \frac{c(w_i, w_j, w_k)}{\sum_{w_l} c(w_l, w_j, w_k)}$$

The number of relations determined in this way can be very large. The upper bound for $P(w_i \mid w_j)$ and $P(w_i \mid w_j w_k)$ are respectively $O(|V|^2)$ and $O(|V|^3)$. However, many relations have very low probabilities and are often noise. As we only consider a subset of strong expansion terms, the relations with low probability are almost never used. Therefore, we set two filtering criteria:

- The biterm in the condition of a relation should be higher than a threshold (10 in our case);
- The probability of a relation should be higher than another threshold (0.0001 in our case).
- One more filtering criterion is mutual information (*MI*), which reflects the relatedness of two terms in their combination (w_j, w_k). To keep a relation $P(w_i \mid w_j w_k)$, we require (w_j, w_k) be a meaningful combination. We use the following pointwise *MI* (Church and Hanks 1989):

$$MI(w_j, w_k) = \log \frac{P(w_j, w_k)}{P(w_j)P(w_k)}$$

We only keep meaningful combinations such that $MI(w_j, w_k) > 0$.

By these filtering criteria, we are able to reduce considerably the number of biterms and triterms. For example, on a collection of about 200MB, with a vocabulary size of about 148K, we selected only about 2.7M useful biterms and about 137M triterms, which remain tractable.

3.3 Probability of Biterms

In LM used in IR, each query term is attributed the same weight. This is equivalent to a uniform probability distribution, i.e.:

$$P(q_i \mid Q) = \frac{1}{|Q|_U}$$

where $|Q|_U$ is the number of unigrams in the query. In CIQE model, we use the same method.

In CDQE, we also need to attribute a probability $P(q_j q_k \mid Q)$, to the biterm (q_j, q_k). Several options are possible.

Uniform probability

This simple approach distributes the probability uniformly among all biterms in the query, i.e.:

$$P(q_j q_k \mid Q) = \frac{1}{|Q|_B}$$

where $|Q|_B$ is the number of biterms in Q.

According to mutual information

In a query, if two words are strongly associated, this also means that their association is more meaningful to the query, thus should be weighted higher. Therefore, a natural way to assign a probability to a biterm in the query is to use mutual information, which denotes the strength of association between two words. We use again the pointwise mutual information $MI(q_j, q_k)$. If it is negative, we consider that the biterm is not meaningful, and is ignored. Therefore, we arrive at the following probability function:

$$P(q_j q_k \mid Q) = \frac{MI(q_j, q_k)}{\sum_{(q_l q_m) \in Q} MI(q_l, q_m)}$$

where $(q_l q_m) \in Q$ means all the meaningful biterms in the query.

Statistical parsing

In (Gao et al., 2002), a statistical parsing approach is used to determine the best combination of translation words for a query. The approach is similar to building a minimal spanning tree, which is also used in (Smeaton and Van Rijsbergen, 1983), to select the strongest term relations that cover the whole query. This approach can also be used in our model to determine the minimal set of the strongest biterms that cover the query.

In our experiments, we tested all the three weighting schemas. It turns out that the best weighting is the one with *MI*. Therefore, in the next section, we will only report the results with the second option.

4. Experimental Evaluation

We evaluate query expansion with different relations on four TREC collections, which are described in Table 1. All documents have been processed in a standard manner: terms are stemmed using Porter stemmer and stopwords are removed. We only use titles of topics as queries, which contain 3.58 words per query on average.

Table 1. TREC collection statistics

Coll.	Description	Size (Mb)	Vocab.	# Doc.	Query
AP	*Associated Press* (1988-89)	491	196,933	164,597	51-100
SJM	*San Jose Mercury News* (1991)	286	146,514	90,257	101-150
WSJ	*Wall Street Journal* (1990-92)	242	121,946	74,520	51-100

In our experiments, the document model remains the same while the query model changes. The document model uses the following Dirichlet smoothing:

$$P(w_i \mid D) = \frac{tf(w_i, D) + \mu P_{ML}(w_i \mid C)}{\mid D \mid_U + \mu}$$

where $tf(w_i, D)$ is the term frequency of w_i in D, $P_{ML}(w_i \mid C)$ is the collection model and μ is the Dirichlet prior, which is set at 1000 following (Zhai and Lafferty, 2001).

There are two other smoothing parameters λ_1, and λ_2 to be determined. In our experiments, we use a simple method to set them: the parameters are tuned empirically using a training collection containing AP1989 documents and queries 101-150. These preliminary tests suggest that the best value of λ_1 and λ_2 (in Equations 1-2) are relatively stable (we will show this later). In the experiments reported below, we will use $\lambda_1 = 0.4$, and $\lambda_2 = 0.3$.

4.1 Experimental Results

The main experimental results are described in Table 2, which reports average precision with different methods as well as the number of relevant documents retrieved. **UM** is the basic unigram model without query expansion (i.e. we use MLE for the query model, while the document model is smoothed with Dirichlet method). **CIQE** is the context-independent query expansion model using unigram relations (Model 1). **CDQE** is the context-dependent query expansion model using biterm relations (Model 2). In the table, we also indicate whether the improvement in average precision obtained is statistically significant (t-test).

Table 2. Avg. precision and Recall

Coll. #Rel.	UM	CIQE	CDQE
AP 6101	0.2767	0.2902 (+5%*)	0.3383 (+22%**) [+17%**]
	3677	3897	4029
SJM 2559	0.2017	0.2225 (+10%**)	0.2448 (+21%**) [+10%*]
	1641	1761	1873
WSJ 2172	0.2373	0.2393 (+1%)	0.2710 (+14%**) [+13%*]
	1588	1626	1737

* and ** indicate that the difference is statistically significant according to t-test: * indicates p<0.05, ** indicates p<0.01; (.) is compared to UM and [.] is compared to CIQE.

CIQE and CDQE vs. UM

It is interesting to observe that query expansion, either by CIQE or CDQE, consistently outperforms the basic unigram model on all the collections. In all the cases except CIQE for WSJ, the improvements in average precision are statistically significant. At the same time, the increases in the number of relevant documents retrieved are also consistent with those in average precision.

The improvement scales obtained with CIQE are relatively small: from 1% to 10%. These correspond to the typical figure using this method.

Comparing **CIQE** and **CDQE**, we can see that context-dependent query expansion (**CDQE**)

always produces better effectiveness than context-independent expansion (**CIQE**). The improvements range between 10% and 17%. All the improvements obtained by CDQE are statistically significant. This result strongly suggests that in general, the context-dependent term relations identify better expansion terms than context-independent unigram relations. This confirms our earlier hypothesis.

Indeed, when we look at the expansion results, we see that the expansion terms suggested by biterm relations are usually better. For example, the (stemmed) expansion terms for the query "*insider trading*" suggested respectively by **CIQE** and **CDQE** are as follows:

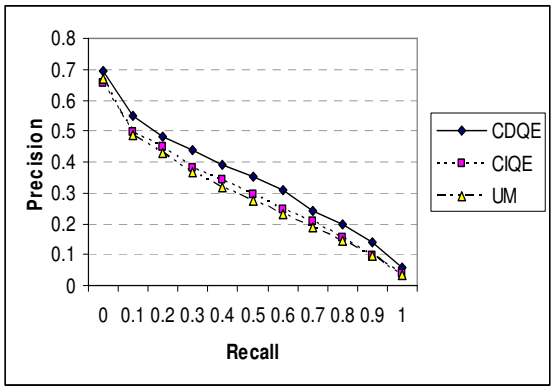
CIQE: stock:0.0141 market:0.0113 US:0.0112 year:0.0102 exchang:0.0101 trade:0.0092 report:0.0082 price:0.0076 dollar:0.0071 1:0.0069 govern:0.0066 state:0.0065 futur:0.0061 million:0.0061 dai:0.0060 offici:0.0059 peopl:0.0059 york:0.0057 issu:0.0057 …

CDQE: **secur**:0.0161 charg:0.0158 stock:0.0137 **scandal**:0.0128 **boeski**:0.0125 inform:0.0119 street:0.0113 wall:0.0112 case:0.0106 year:0.0090 million:0.0086 **investig**:0.0082 exchang:0.0080 govern:0.0077 **sec**:0.0077 **drexel**:0.0075 **fraud**:0.0071 law:0.0063 ivan:0.0060 …

We can see that in general, the terms suggested by **CDQE** are much more relevant. In particular, it has been able to suggest "**boeski**" (Boesky) who is involved in an insider trading scandal. Several other terms are also highly relevant, such as **scandal, investing, sec, drexel, fraud**, etc.

The addition of these new terms does not only improve recall. Precision of top-ranked documents is also improved. This can be seen in Figure 1 where we compare the full precision-recall curve for the AP collection for the three models. We can see that at all the recall levels, the precision values always follow the following order: CDQE > UM. The same observation is also made on the other collections. This shows that the CDQE method does not increase recall to the detriment of precision, but both of them. In contrast, CIQE increases precision at all but 0.0 recall points: the precision at the 0.0 recall point is 0.6565 for CIQE and 0.6699 for UM. This shows that CIQE can slightly deteriorate the top-ranked few documents.

Figure 1. Comparison of three models on AP



CDQE vs. Pseudo-relevance feedback

Pseudo-relevance feedback is widely considered to be an effective query expansion method. In many previous experiments, it produced very good results. The mixture model (Zhai and Lafferty, 2001) is a representative and effective method to implement pseudo-relevance feedback: It uses a set of feedback documents to smooth the original query model. Compared to the mixture model, our CDQE method is also more effective: By manually tuning the parameters of the mixture model to their best, we obtained the average precisions of 0.3171, 0.2393 and 0.2565 respectively for AP, SJM and WSJ collections. These values are lower than those obtained with CDQE, which has not been heavily tuned.

For the same query "*insider trading*", the mixture model determines the following expansion terms:

Mixture: stock:0.0259256 secur:0.0229553 market:0.0157057 sec:0.013992 inform:0.011658 firm:0.0110419 exchang:0.0100346 law:0.00827076 bill:0.007996 case:0.00764544 profit:0.00672575 investor:0.00662856 japan:0.00625859 compani:0.00609675 commiss:0.0059618 foreign:0.00582441 bank:0.00572947 investig:0.00572276

We can see that some of these terms overlap with those suggested by biterm relations. However, interesting words such as *boeski*, *drexel* and *scandal* are not suggested.

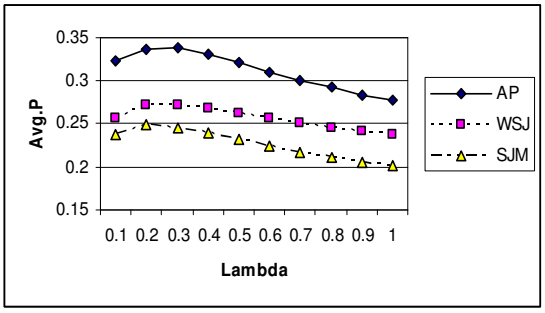
The above comparison shows that our method outperforms the state-of-the-art methods of query expansion developed so far.

4.2 Effect of the Smoothing Parameter

In the previous experiments, we have fixed the smoothing parameters. In this series of tests, we

analyze the effect of this smoothing parameter on retrieval effectiveness. The following figure shows the change of average precision (AvgP) using **CDQE** (Model 2) along with the change of the parameter λ_2 (UM is equivalent to $\lambda_2 = 1$).

Figure 2. Effectiveness w.r.t. λ_2



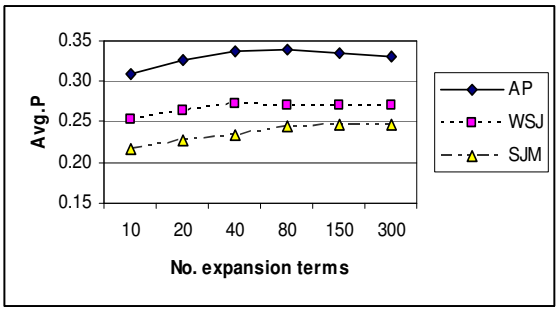
We can see that for all the three collections, the effectiveness is good when the parameter is set in the range of 0.1-0.5. The best value for different collections remains stable: 0.2-0.3.

The effect of λ_1 on Model 1 is slightly different, but we observe the same trend.

4.3 Number of Expansion Terms

In the previous tests, we limit the number of expansion terms to 80. When different numbers of expansion terms are used, we obtain different effectiveness measures. The following figure shows the variation of average precision (AvgP) with different numbers of expansion terms, using **CDQE** method.

Figure 3. Effectiveness w.r.t. #expansion terms



We can see that when more expansion terms are added, the effectiveness does not always increase. In general, a number around 80 will produce good results. In some cases, even if better effectiveness can be obtained with more expansion terms, the retrieval time is also longer. The number 80 seems to produce a good compromise between effectiveness and retrieval speed: the retrieval time remains less than 1 sec. per query.

4.4 Suitability of Relations Across Collections

In many real applications (e.g. Web search), we do not have a static document collection from which relations can be extracted. The question is whether it is possible and beneficial to extract relations from one text collection and use them to retrieve documents in another text collection. Our intuition is that this is possible because the relations (especially context-dependent relations) encode general knowledge, which can be applied to a different collection. In order to show this, we extracted term relations from each collection, and applied them on other collections. The following tables show the effectiveness produced using respectively unigram and bi-term relations.

Table 3. Cross-utilization of relations

Coll. \ Rel.	Unigram relation			Biterm relation		
	AP	**SJM**	**WSJ**	**AP**	**SJM**	**WSJ**
AP	**0.2902**	0.2803	0.2793	**0.3383**	0.3057	0.2987
SJM	**0.2271**	0.2225	0.2267	0.2424	0.2448	**0.2453**
WSJ	**0.2541**	0.2445	0.2393	**0.2816**	0.2636	0.2710

From this table, we can observe that relations extracted from any collection are useful to some degree: they all outperform **UM** (see Table 2). In particular, the relations extracted from AP are the best for almost all the collections. This can be explained by the larger size and wider coverage of the AP collection. This suggests that we do not necessarily need to extract term relations from the same text collection on which retrieval is performed. It is possible to extract relations from a large text collection, and apply them to other collections. This opens the door to the possibility of constructing a general relation base for various document collections.

5. Related Work

Co-occurrence analysis is a common method to determine term relations. The previous studies have been limited to relations between two words, which we called unigram relations. This expansion approach has been integrated both in traditional retrieval models (Jing and Croft, 1994) and in LM (Berger and Lafferty 1999). As we observed, this type of relation will introduce much noise into the query, leading to unstable effectiveness.

Several other studies tried to filter out noise expansion (or translation) terms by considering the relations between them (Gao et al., 2002;

Jang et al. 1999; Qiu and Frei, 1993; Bai et al. 2005). However, this is insufficient to detect all the noise. The key issue is the ambiguity of relations due to the lack of context information in the relations. In this paper, we proposed a method to add some context information into relations.

(Lin, 1997) also tries to solve word ambiguity by adding syntactic dependency as context. However, our approach does not require determining syntactic dependency. The principle of our approach is more similar to (Yarowsky, 1995). Compared to this latter, our approach is less demanding: we do not need to identify manually the exact word senses and seed context words. The process is fully automatic. This simplification is made possible due to the requirement for IR: only in-context related words are required, but not the exact senses.

Our work is also related to (Smadja and McKeown, 1996), which tries to determine the translation of collocations. Term combinations or biterms we used can be viewed as collocations. Again, there is much less constraint for our related terms than translations in (Smadja and McKeown, 1996).

6. Conclusions

In many NLP applications such as IR, we need to determine relations between terms. In most previous studies, one tries to determine the related terms to one single term (word). This makes the resulting relations ambiguous. Although several approaches have been proposed to remove afterwards some of the inappropriate terms, this only affects part of the noise, and much still remains. In this paper, we argue that the solution to this problem lies in the addition of context information in the relations between terms. We proposed to add another word in the condition of the relations so as to help constrain the context of application. Our experiments confirm that this addition of limited context information can indeed improve the quality of term relations and query expansion in IR.

In this paper, we only compared biterm relations and unigram relations, the general method can be extended to triterm relations or more complex relations, provided that they can be extracted efficiently.

This paper only investigated the utilization of context-dependent relations in IR. These relations can be applied in many other tasks, such as machine translation, word sense disambiguation / discrimination, and so on. These are some interesting research work in the future.

References

Bai, J., Song, D., Bruza, P., Nie, J. Y. and Cao, G. 2005. Query expansion using term relationships in language models for information retrieval, *ACM CIKM*, pp. 688-695.

Berger, A. and Lafferty, J. 1999. Information retrieval as statistical translation. *ACM SIGIR*, pp. 222-229.

Church, K. W. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. *ACL*, Vol. 16, pp. 22-29.

Gao, J., Nie, J.Y., He, H, Chen, W., Zhou, M. 2002. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependency relations. *ACM SIGIR,* pp. 11-15.

Jang, M. G., Myaeng, S. H., and Park, S. Y. 1999. Using mutual information to resolve query translation ambiguities and query term weighting. *ACL*, pp. 223-229.

Jing, Y. and Croft, W.B. 1994. An association thesaurus for information retrieval. *RIAO*, pp. 146-160.

Lin, D. 1997. Using syntactic dependency as local context to resolve word sense ambiguity, *ACL,* pp. 64-71.

Peat, H.J. and Willett, P. 1991. The limitations of term co-occurrence data for query expansion in document retrieval systems. *JASIS,* 42(5): 378-383.

Ponte, J. and Croft, W.B. 1998. A language modeling approach to information retrieval. *ACM SIGIR*, pp. 275-281.

Qiu, Y. and Frei, H.P. 1993. Concept based query expansion. *ACM SIGIR*, pp.160-169.

Schütze, H. and Pedersen J.O. 1997. A cooccurrence-based thesaurus and two applications to information retrieval, *Information Processing and Management,* 33(3): 307-318.

Smeaton, A. F. and Van Rijsbergen, C. J. 1983. The retrieval effects of query expansion on a feedback document retrieval system. *Computer Journal*, 26(3): 239-246.

Smadja, F., McKeown, K.R., 1996. Translating collocations for bilingual lexicons: A statistical approach, *Computational Linguistics*, 22(1): 1-38.

Srikanth, M. and Srihari, R. 2002. Biterm language models for document retrieval. *ACM SIGIR*, pp. 425-426

Voorhees, E. 1994. Query expansion using lexical-semantic relations. *ACM SIGIR*, pp. 61-69.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *ACL*, pp. 189-196.

Zhai, C. and Lafferty, J. 2001. Model-based feedback in the language modeling approach to information retrieval. *ACM SIGIR*, pp. 403-410.