Query Expansion Using Term Relationships in Language Models for Information Retrieval

Jing Bai[1], Dawei Song[2], Peter Bruza[3], Jian-Yun Nie[1], Guihong Cao[1]

[1] DIRO, University of Montreal
CP. 6128 succ. Centre-ville,
Montreal, Quebec, H3C 3J7 Canada
{baijing, nie,
caogui}@iro.umontreal.ca

[2] Knowledge Media Institute,
The Open University, Walton Hall,
Milton Keynes, MK7 6AA, UK
dawei_song2005@hotmail.com

[3] Distributed Systems Technology Centre
University of Queensland, QLD 4072, Australia
bruza@dstc.edu.au

ABSTRACT

Language Modeling (LM) has been successfully applied to Information Retrieval (IR). However, most of the existing LM approaches only rely on term occurrences in documents, queries and document collections. In traditional unigram based models, terms (or words) are usually considered to be independent. In some recent studies, dependence models have been proposed to incorporate term relationships into LM, so that links can be created between words in the same sentence, and term relationships (e.g. synonymy) can be used to expand the document model. In this study, we further extend this family of dependence models in the following two ways: (1) Term relationships are used to expand query model instead of document model, so that query expansion process can be naturally implemented; (2) We exploit more sophisticated inferential relationships extracted with Information Flow (IF). Information flow relationships are not simply pairwise term relationships as those used in previous studies, but are between a set of terms and another term. They allow for context-dependent query expansion. Our experiments conducted on TREC collections show that we can obtain large and significant improvements with our approach. This study shows that LM is an appropriate framework to implement effective query expansion.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval – retrieval models.

General Terms: Experimentation, Performance

Keywords: Language model, Term relationships, Information flow, Query expansion

1. INTRODUCTION

Language Modeling (LM) is an approach used in many recent studies in IR. It not only produces promising experimental results (comparable to the best IR systems), but also provides a solid theoretical setting. However, the classical LM approaches usually assume independence between indexing units, which are unigrams or bigrams. In reality, a word may be related to other words. An example is the synonymy relationship. Such relationships between terms should be properly integrated into LM.

Some recent approaches try to extend the existing LM by incorporating term relationships or dependencies [1, 4, 6]. Term relationships are considered in the following two perspectives:

(1) One may create relationships or links among document terms or among query terms. In such a way, a sentence (either in a query or in a document) is interpreted not only as a set of words, but also as a set of relationships or links among the words. Under such an interpretation, for a document to be retrieved, it has to contain not only the words required in the query as in the classical LM, but also the additional links required by the query. Therefore, term relationships are used to enhance document and query representations in this perspective. This is the idea of dependence model implemented in [6].

(2) Term relationships can also be considered between query terms and document terms, so that indirect correspondence between document and query can be inferred during query evaluation. This is what is done in [1, 4]: If a document does not contain the same terms as the query does, but contains some related terms, it can still be retrieved by using the term relationships. In both [1] and [4], term relationships are used to expand the document model, so that the probability of the related terms in the document will be increased.

Our work also aims to exploit relationships between query terms and documents terms in the context of the second perspective as [1, 4]. However, there are two differences:

(1) The previous work often aims to improve the document model in order to increase probabilities of related terms in a document. This can be regarded as a document expansion approach. Our work aims to use term relationships in query expansion within the LM framework.

(2) There are not many large linguistic resources such as Wordnet available for IR. Therefore, a question that one may raise is, beside co-occurrence relationships, can we extract other types of relationship from data that can be incorporated into LM? In this paper, we exploit inferential term relationships extracted by using a more sophisticated approach, namely Information Flow (IF). Unlike the traditionally used pairwise term relationships, these relationships are context-dependent, in the sense that they are between a set of terms and another term (e.g. (Java, computer) → programming). This would help reduce the inappropriate applications of the relationships in wrong contexts (when ambiguity arises). The information flow model has produced encouraging results when employed in concert with the classical

CIKM'05, October 31-November 5, 2005, Bremen, Germany.

vector space model [2, 15]. In this paper, we integrate relationships computed by information flow into a LM.

It will be shown that the idea of query expansion can be integrated into LM in a straightforward way – query expansion via term relationship can be seen as a new smoothing of query model. Query expansion in LM has also been investigated in several previous studies [8, 9, 11, 12, 18]. However, these latter all rely on feedback documents to enhance the original query model. In our work, we use explicit relationships between terms for query expansion, similarly to traditional query expansion approaches. Our experiments will show that we can obtain improvements by expanding the query model using both co-occurrence relationships and IF relationships, but IF relationships make a much larger contribution to it.

This paper is organized as follows. The next section briefly describes the existing LM approaches. In Section 3, we first give an overview on query expansion via LM, and then describe our approach of integrating term relationships into LM for query expansion. Section 4 presents the method of deriving inferential term relationships. Our experimental set-up and the empirical results on the TREC data set are presented in Section 5. Finally, section 6 concludes the paper and highlights some future directions.

2. EXISTING LANGUAGE MODELS

2.1 Classical LM

The basic idea of LM for IR is to compute the conditional probability $P(Q|D)$, i.e., the probability of generating a query Q given the observation of a document D [5]. Documents are then ranked in descending order of this probability.

Assuming that words in the query are independent, we have a general unigram model formulated as follows:

$$P(Q \mid D) = \prod_{q_i \in Q} P(q_i \mid D) \qquad (1)$$

where q_i is a term (unigram in this paper) in the query.

Another popular formulation of LM in IR is the KL-divergence [5]. In this case, a document model is estimated, so is a query model. The score is then determined by KL-divergence between the two models. More specifically:

$$\begin{aligned} Score(Q,D) &= \sum_{t_i \in V} P(t_i \mid Q) \times \log P(t_i \mid D) \\ &\overset{rank}{\propto} -KL(M_Q, M_D) \end{aligned} \qquad (2)$$

where V is the vocabulary of the collection, M_Q and M_D are respectively language models for Q and D. Note that in previous studies, $P(t_i \mid Q)$ is often directly determined by Maximum Likelihood (ML) estimation, i.e., $P(t_i \mid Q) = tf(t_i, Q) / \sum_{t_i} tf(t_i, Q)$, where $tf(t_i,Q)$ is the term frequency of t_i in Q. In this way, the sum can be restricted to query terms only, i.e.,

$$Score(Q,D) = \sum_{q_i \in Q} P(q_i \mid Q) \times \log P(q_i \mid D)$$

In practice, this restriction also has the advantage of reducing the complexity of the query evaluation process.

We can observe that both formulas still require the query terms to appear in a document for the latter to be retrieved. With smoothing of the document model, one can increase the probability $P(q_i \mid D)$ of a term q_i absent in D from zero to a small value, thus making it possible for a document not containing some of the query terms to be retrieved. It is important to note that smoothing increases the probabilities of all the non-occurring terms in the document uniformly or proportionally to the term distribution in the whole collection. No distinction is made between the terms that are really related to the document terms and those that are not related.

In fact, some terms are strongly related to others. For example, suppose that the term "algorithm" appears in a document. The probability that the document satisfies a query on "computer" is much higher than that for a query on, say, "elephant". Therefore, if both terms are absent from a document, then the increase of the probabilities of "computer" and "elephant" in the document should not be equal or simply proportional to their distribution in the collection. Instead, it should be a function based on the strength of their relationships to the document term "algorithm". This is the idea of [4] which incorporates term relationships in document language models.

2.2 Dependence LM for document expansion

Now let us examine how a query term is satisfied in a document (i.e. the document is relevant to the query term). On one hand, if the query term is contained in the document, it is certainly satisfied by the document to some degree. On the other hand, however, a query term not appearing in a document does not necessarily mean that the document is not relevant. There may be some other terms in the document that are strongly related to the given query term, for example, synonyms. In this second case, the document can still be judged relevant to some extent through the term relationships. Taking both cases into account, the probability $P(q_i \mid D)$ can be formulated as follows:

$$P(q_i \mid D) = P(q_i, \theta_U \mid D) + P(q_i, \theta_R \mid D)$$

where θ_U and θ_R represent respectively the independent unigram model and the model with term relationships. This expression means that the satisfaction of a query term can be either a direct satisfaction or a satisfaction through term relationships. Starting with this development, we can further derive:

$$P(q_i \mid D) = P(q_i \mid \theta_U, D) P(\theta_U \mid D) + P(q_i \mid \theta_R, D) P(\theta_R \mid D)$$

In this formula, $P(\theta_U \mid D)$ and $P(\theta_R \mid D)$ determines the probability of selecting each of the models, given a document. $P(q_i \mid \theta_U, D)$ can be estimated by $P(q_i \mid D)$ in the unigram model, and $P(q_i \mid \theta_R, D)$ can be estimated by considering any kind of term relationships. In [4], co-occurrences, as well as relationships in Wordnet, are used to estimate $P(q_i \mid \theta_R, D)$. Alternatively, the translation relationships proposed in [1] can also be used.

If we limit our consideration to the above two models only, we have $P(\theta_U \mid D) + P(\theta_R \mid D) = 1$. A further simplification is to assume equal $P(\theta_U \mid D)$ for different D. The above formula can then be viewed as another variation of smoothing between a unigram model and a dependence model:

$$P(Q \mid D) = \prod_{q_i \in Q} [\lambda P(q_i \mid \theta_U, D) + (1-\lambda) P(q_i \mid \theta_R, D)] \qquad (3)$$

where $\lambda = P(\theta_U \mid D)$ is a mixture parameter.

We can see here that the effect of integrating term relationships is to create a new document model $P(.|D)$, which is smoothed by a relation model (the second term of the above formula). We can therefore term the approach "document expansion".

2.3 Query expansion with feedback

The converse of "document expansion" is "query expansion", the goal of which is to obtain a better query description. In terms of LM, it aims to build a better query model.

Several previous studies have investigated query expansion via pseudo-relevance feedback. [12] selects the terms that have high probability in the feedback documents, but low probability in the collection, as expansion terms. These terms are integrated into a new query model. [11] uses a similar approach.

More recent studies try to exploit feedback documents in model-based approaches. Basically, feedback documents are used to create different language models, which are considered to be samples for relevance [9] or for a new query model [8, 18]. The original query terms are used to focus the selection among these sample models. The selected sample models are then used to form a new query/relevance model.

Although all the above approaches also aim at building a new query model, which is expanded from the original query in a certain way, we notice that these approaches basically make use of a new term distribution within a subset of documents and within the collection. No term relationships are explicitly used, which, in our opinion, is necessary to determine $P(t_i|Q)$. Our approach aims precisely to make use of term relationships in query expansion.

3. QUERY EXPANSION WITH TERM RELATIONSHIPS

With respect to formula (2), query expansion consists of finding a better way of estimating $P(t_i|Q)$, so that not only the terms expressed in the query will have a non-zero probability, but also other related terms. While this idea is intuitive and appealing, it has not been fully incorporated into LM framework.

Classical smoothing techniques, by combining the collection model or other term distributions, can only redress this issue to some extent: One can arrive at a smoothed query model in which more terms will receive non-zero probabilities, which are thus taken into account in query evaluation. However, as we already stated, if we only rely on term frequency redistribution via smoothing, the effect of term relationships is simply ignored. For example, it is likely that for a query on "computer", the new term "algorithm" may not necessarily receive a higher probability than the term "elephant" after the redistribution. As a result, the term "elephant" will have equal or even higher importance to "algorithm" in the evaluation of the query on "computer", which is counter-intuitive. In this case, it is more reasonable to assign a higher probability to the term "algorithm" due to the relationship between "algorithm" and "computer". In addition, a naïve utilization of smoothing by a collection model results in a large query model (having all terms with non-zero probability), thereby increasing the cost of computing the matching function between query and document models.

Our alternative solution is to smooth the original query model $P_{ML}(t_i|Q)$ by another probability function $P_R(t_i|Q)$ determined according to some explicit term relationships:

$$P(t_i \mid Q) = \lambda P_{ML}(t_i \mid Q) + (1-\lambda) P_R(t_i \mid Q) \qquad (4)$$

where λ is a mixture parameter as in formula (3). This formula expresses the main idea of our approach. It is similar to that proposed in [18]. However, as one will see, our expansion is based on explicit term relationships instead of pseudo-relevance feedback.

Putting this into formula (2) using KL-divergence, we obtain the following formula:

$$\begin{aligned} Score(Q,D) &= \sum_{t_i \in V} P(t_i \mid Q) \times \log P(t_i \mid D) \\ &= \sum_{t_i \in V} [\lambda P_{ML}(t_i \mid Q) + (1-\lambda) P_R(t_i \mid Q)] \times \log P(t_i \mid D) \\ &= \lambda \sum_{q_i \in Q} P_{ML}(q_i \mid Q) \times \log P(q_i \mid D) + (1-\lambda) \sum_{t_i \in V} P_R(t_i \mid Q) \times \log P(t_i \mid D) \end{aligned} \qquad (5)$$

Notice that the first term is a sum over the query words (instead of all the vocabulary). This is because $P_{ML}(t_i \mid Q) = 0$ for $t_i \notin Q$. The second term corresponds to the classical query expansion process, in which some (new) terms t_i related to the query Q are determined, and their probabilities in the document model are used in query evaluation.

Similarly to the earlier case on document expansion, there are several possible ways to determine $P_R(t_i|Q)$. In this study, we investigate two following ones:

- One can use co-occurrence information to generate statistical relationships between terms, and use these relationships to "smooth" the query model;
- One can also use some other types of knowledge, such as Wordnet or other knowledge extracted from document collection.

Whatever the relationships used, it is important to restrict the number of terms t_i to be considered in the second term of formula (5) in order to render the approach more computationally efficient. Therefore, some selection or filtering of terms according to their $P_R(t_i|Q)$ should be done. This selection does not raise the same zero-probability problem as for document model, because with a zero-probability for $P_R(t_i|Q)$ in formula (5), the whole score will not be $-\infty$.

4. USING TERM RELATIONSHIPS

4.1 Term relationships from co-occurrence counts

Term co-occurrences have been used to derive term relationships in a number of studies [14]. Typically, one observes the frequency of term co-occurrences within a certain context, which can be the whole document, passage, or a window of fixed length. Then the strength (or probability) of term relationship is calculated as follows:

$$P_{co}(t_2 \mid t_1) = \frac{f(t_1, t_2)}{\sum_{t_i} f(t_1, t_i)} \qquad (6)$$

where $f(t_1, t_2)$ is the frequency of co-occurrences of t_1 and t_2.

4.2 Co-occurrence in HAL space

HAL (Hyperspace Analogue to Language) is a cognitively motivated and validated semantic space model for deriving term co-occurrence relationships [3, 10]. What HAL does is to generate a word-by-word co-occurrence matrix from a large text corpus via a l-sized sliding window: All the words occurring within the window are considered as co-occurring with each other. By moving the window across the text corpus, an accumulated co-

occurrence matrix for all the words in a certain vocabulary is produced. The strength of association between two words is inversely proportional to their distance. This idea is similar to the decaying factor according to distance used in [7].

An example showing the HAL space for the text "the effects of pollution on the population" using a 5-word moving window (l=5) is depicted in Table 1.

Table 1. Example of a HAL space

	the	effects	of	pollution	on	population
the	1	2	3	4	5	
effects	5					
of	4	5				
pollution	3	4	5			
on	2	3	4	5		
population	5	1	2	3	4	

The original HAL space is direction sensitive: The co-occurrence information preceding and following a word are recorded separately by the row and column vectors. However, for the purpose of deriving term relationships in IR, word order does not seem to be important. Therefore, the HAL vector of a word is represented by adding up its row and column vectors. For a given word, the dimensions in its HAL vector whose weights are higher than a threshold (set at the mean positive weight in our experiments) are called "quality properties" of the word.

To fit in the LM framework, a probabilistic HAL space can be estimated by normalizing a HAL vector by the sum of all the dimension weights:

$$P_{HAL}(t_2 \mid t_1) = \frac{HAL(t_2 \mid t_1)}{\sum_{t_i} HAL(t_i \mid t_1)}$$

where $HAL(t_2|t_1)$ is the weight of t_2 in the HAL vector of t_1.

Below are the probability values we obtain for the simple example:

pollution = {effects:0.17, of:0.21, on: 0.21, population: 0.12, the:0.29}

This example demonstrates how a word is represented as a weighted vector whose dimensions comprise other words. The weights represent the strengths of associations between "pollution" and other words seen in the context of the sliding window: The higher the weight of a word, the more it has lexically co-occurred with "pollution" in the same context(s).

From a large corpus, the vector derived contains much noise. In order to reduce noise, only the dimensions with weights above the mean are kept and then normalized so that they sum to 1. Thus we can consider the weight $P_{HAL}(t_2 \mid t_1)$ as an alternative probability function for $P_{co}(t_2 \mid t_1)$.

Different words can be combined to form more complex concepts like "space program". A vector is also obtained for this latter by combining the HAL vectors of the individual terms. A simple method is to add the vectors of the terms. In this article, however, we employ a more sophisticated concept combination heuristic [2]. It can be envisaged as a weighted addition of underlying vectors paralleling the intuition that in a given concept combination, some terms are more dominant than others. For example, the combination "space program" is more "space-*ish* than "program-*ish*". Dominance is determined by the specificity of the term, that is, dominance is assumed to correlate with the *idf* of a term. Space restrictions preclude a more detailed description of the concept combination heuristic, but by way of illustration, we have the following vector for the concept combination "space program":

{U.S.:0.11 aboard:0.04 administration:0.17 aeronautics:0.15 agency:0.15 air:0.04 america:0.06 american:0.05 astronauts:0.04 based:0.09 billion:0.07 budget:0.04 bush:0.07 center:0.18 challenger:0.04 commercial:0.04 council:0.06 defense:0.07 development:0.05 director:0.03 discovery:0.03 earth:0.03 european:0.04 exploration:0.08 flight:0.13 grant:0.07 house:0.03 johnson:0.04 kennedy:0.05 launch:0.08 launched:0.04 manned:0.10 marshall:0.03 million:0.05 mir:0.04 missile:0.04 mission:0.04 nasa:0.12 national:0.23 new:0.09 officials:0.06 president:0.06 probe:0.04 program:0.40 programs:0.05 quayle:0.04 research:0.07 rocket:0.08 science:0.08 shuttle:0.37 soviet:0.17 space:0.38 star:0.04 station:0.33 technology:0.06 unmanned:0.04 wars:0.03 work:0.03}

4.3 Information Flow (IF)

Information flow is a mechanism developed to do information inference [15]. We say that there is an information flow from a set of terms (or information items) $t_1, \ldots, t_k$ to another term t_j if the former entails, or "suggests", to some degree, the latter. This is denoted as $t_1, \ldots, t_k \mid\!- t_j$. The terms $t_1, \ldots, t_k$ are referred to as the "premise".

In the previous work, [2] developed a heuristic way to extract information flows. Their approach can be summarized as follows:

- The initial HAL space is filtered so that for each term, only strong co-occurring terms are kept as "quality properties" of the term;
- The degree of information flow from $t_1, \ldots, t_k$ to t_j is defined as follows:

$$t_1, \ldots, t_k \mid\!- t_j \text{ iff } \text{degree}(\bigoplus_{1 \le i \le k} c_i \lhd c_j) > \delta$$

$$\text{degree}(\bigoplus_{1 \le i \le k} c_i \lhd c_j) = \frac{\sum_{p_l \in (QP(\oplus c_i) \wedge QP(c_j))} w_{c_i p_l}}{\sum_{p_l \in QP(\bigoplus_{1 \le i \le k} c_i)} w_{c_i p_l}}$$

where c_i denotes the HAL vector for term t_i, $\bigoplus_{1 \le i \le k} c_i$ denotes the concept combination vector of individual vectors c_i, $w_{c_i p_l}$ denotes the weight of component p_l in vector c_i, $QP(.)$ refers the set of quality properties of the vector in question, and δ a threshold which determines the strength of information flow necessary to sanction the associated inference. In our case, this threshold plays no role as terms will be ranked according to the degree of information flow from the premise being a set of query terms. The top raked terms will then be used to prime the query model. More details about this come shortly.

Information flow is a normalized score which essentially measures how many of the quality properties of the source vector are also quality properties of the target vector. The more such quality properties, the higher the information flow. Maximal information flow is achieved when all quality properties of the source vector "map" into the target. For example, some top ranked information flows from "space program" (TREC query 011) are listed as follows (where the numbers are degrees):

space, program **|-**

{program:1.00 space:1.00 nasa:0.97 new:0.97 U.S.:0.96 agency:0.95 shuttle:0.95 national:0.95 soviet:0.95 president:0.94 bush:0.94 million:0.94 launch:0.93 **called:0.93** **thursday:0.93** research:0.92

administration:0.92 flight:0.92 rocket:0.92 defense:0.91 **Friday:0.91 project:0.91 system:0.91** mission:0.91 work:0.90 launched:0.90 officials:0.90 station:0.89 **long:0.88 announced:0.88** science:0.88 **scheduled:0.87 reagan:0.87** director:0.87 programs:0.87 air:0.87 **put:0.87** center:0.87 billion:0.87 aeronautics:0.87 **satellite:0.87 force:0.86 news:0.86 wednesday:0.86** technology:0.86 american:0.86 budget:0.86 **states:0.86 back:0.85 office:0.85 monday:0.85 plan:0.85 people:0.85** manned:0.85 **satellites:0.85** …}

Notice that the bolded terms in the above example, such as "satellite", "pentagon", "scientists", etc., are absent from the vector for "space program". These new terms appear in the above vector because they share many of the context words with "space program". Some of the added terms (e.g. satellite) are indeed closely related to "space program". This example shows the possible benefit of IF.

Note that the degrees of information flow in the above example are in range of (0,1] rather than probabilities. To fit in the LM framework, the probability of information flow relationships can be computed as:

$$P_{IF}(t_2 \mid t_1) = \frac{\text{degree}(c_1 \lhd c_2)}{\sum_{t_k \in Vocabulary} \text{degree}(c_1 \lhd c_k)}$$

In comparison with the classical co-occurrence relationships, IF have the following unique characteristics:

- IF is constructed on a filtered HAL space. Therefore, a lot of statistical noise has been removed before IF is extracted.
- Information flow computation allows for "genuine" inferences. In the example of "space program", we see that some new related terms can be inferred.
- IF is not always a relationship between pairs of terms. Rather, it can be a relationship between a set of terms and a new term. By using a set of terms as premise, one is able to account for context-dependent relationships. For example, while in general, "program" can entail "computer" (in particular, for a computer-related corpus), "space program" should not entail the same term, but rather "satellite". We can see that IF relationships may encode more complex, context-dependent relationships.
- The use of term combination allows us not to be limited to syntactically valid phrases only. It is a more flexible way of deriving information flows from any arbitrary composition of related terms.

4.4 Query expansion using term relationships

Now we explain how we can use different term relationships to define $P_R(t_i \mid Q)$ in formula (5). We consider two cases: (1) using relationships derived from raw co-occurrence data (or HAL space); (2) using inferential relationships of IF.

Using co-occurrence or HAL relationships

Assume a set of term relationships defined between pairs of terms with probability $P_{HAL}(t_i \mid t_j)$. We can estimate $P_R(t_i \mid Q)$ as follows:

$$P_R(t_i \mid Q) = P_{HAL}(t_i \mid Q) = \sum_{q_j \in Q} P_{HAL}(t_i \mid q_j) \times P(q_j \mid Q) \quad (7)$$

Putting it into formula (5), we obtain:

$$Score(Q,D) = \lambda_{HAL} \sum_{q_i \in Q} P_{ML}(q_i \mid Q) \times \log P(q_i \mid D) + (1-\lambda_{HAL}) \sum_{t_i \in V} \sum_{q_j \in Q} P_{HAL}(t_i \mid q_j) \times P(q_j \mid Q) \times \log P(t_i \mid D)$$

As noted earlier, in practice, it is computationally inefficient to prime query models with positive probabilities for all terms in the vocabulary. Some selection is warranted. One way to do this is to limit to a reasonable number of terms during the expansion (for example, a set of strongest relationships). Assume that a set E of term relationships is selected. We then have the following approximation:

$$Score(Q,D) \approx \lambda_{HAL} \sum_{q_i \in Q} P_{ML}(q_i \mid Q) \times \log P(q_i \mid D) + (1-\lambda_{HAL}) \sum_{q_j \in Q \wedge R(t_i,q_j) \in E} P_{HAL}(t_i \mid q_j) \times P(q_j \mid Q) \times \log P(t_i \mid D) \quad (8)$$

where $R(t_i, q_j) \in E$ means that the term relationship between t_i and q_j is selected.

This formula corresponds to the approach using pairwise term relationships for query expansion. One may observe some similarity between formula (7) and the translation model proposed in [1]. However, our formula is used to expand query model, while that of [1] is used to expand document model.

Using information flow

Assume that we have extracted a set of IF relationships. Then $P_R(t_i \mid Q)$ can be developed as follows:

$$P_R(t_i \mid Q) = P_{IF}(t_i \mid Q) = \sum_{Q_j \subseteq Q} P_{IF}(t_i \mid Q_j) \times P(Q_j \mid Q)$$

where Q_j is a single query term, or a group of query terms, corresponding to the premise of an IF relationship. An important difference from the previous formula is that term relationships are no longer pairwise. The idea of using a set of terms as the premise of an IF relation is similar to that of [13]. In [13], it has been shown that the best approach to query expansion is to determine the expansion terms not according to their relationships to individual query terms, but to the whole query. In our formulation, Q_j expresses a set of query terms that are necessary to determine an appropriate expansion term. So our above formula is an implementation of the idea of [13] in the LM framework.

$P(Q_j|Q)$ is another component to be determined. A possible way is to determine it according to the probabilities of terms composing Q_j in the query. In our experiments, however, we will take a simpler manner: $P(Q_j|Q)$ is assigned an equal value, i.e., $1/|Q'|$ where Q' is the number of terms in the expanded query. This simple solution does not have any impact in our experiments, because our queries are very short, and usually all the original query terms are included in the premise of the IF relationships used.

Formula (5) now becomes:

$$Score(Q,D) = \lambda_{IF} \sum_{q_i \in Q} P_{ML}(q_i \mid Q) \times \log P(q_i \mid D) + (1-\lambda_{IF}) \sum_{t_i \in V} \sum_{Q_j \subseteq Q} P_{IF}(t_i \mid Q_j) \times P(Q_j \mid Q) \times \log P(t_i \mid D)$$

As previously, it is necessary to limit the number of term relationships used in query expansion. Assume that we have selected a set E of the strongest IF relationships. The following approximation can then be made:

$$Score(Q,D) \approx \lambda_{IF} \sum_{q_i \in Q} P_{ML}(q_i \mid Q) \times \log P(q_i \mid D) + (1-\lambda_{IF}) \sum_{Q_j \subseteq Q \wedge R(t_i,Q_j) \in E} P_{IF}(t_i \mid Q_j) \times P(Q_j \mid Q) \times \log P(t_i \mid D) \quad (9)$$

Note that our discussions above focus on query model. For the document model, one has to use a smoothing method as proposed in other studies [18]. We will test several smoothing methods in our experiments.

5. EXPERIMENTAL EVALUATION

5.1 Experimental setup

We evaluate our model described in the previous sections using TREC collections – AP (Associated Press). Some statistics are shown in Table 2. All documents have been processed in a standard manner: Terms are stemmed using the Porter stemmer and stop words are removed. The experiments reported here use the AP89 collection (TREC disk 1) for topics 1-50, and the AP 88&89 collection (TREC disks 1 and 2) with TREC topics 101-150 and 151-200. Only the titles of the topics are used as queries.

Table 2. Text collection statistics

Corpus	# Doc.	Size (Mb)	Vocab.	Query	Q. length
AP89	84, 678	262	137, 728	1-50	3.2
AP88-89	164, 597	507	249, 453	101-150	3.6
AP88-89	164, 597	507	249, 453	151-200	4.3

We test the classical LM and three query expansion models as follows:

Basic LM: As reference models, we use unigram models with different smoothing techniques proposed in [17].

HAL-based query expansion: This model is used to investigate whether using co-occurrence information via HAL in query expansion can bring some improvements. In this model, the HAL spaces are constructed from both collections using a window size of 8 words (i.e. l = 8). For expanding the query model, we select the top 85 highly weighted dimensions (quality properties) in the vector representation of the combination of query terms, which is derived via the concept combination heuristic. Both the window size and the number of vector dimensions are set empirically, and they have led to good performance in a previous study [2].

Global IF-based query expansion: This test is to investigate whether information flow analysis contributes positively to query model derivation. The top 85 information flows extracted from the whole document collection are used to expand the query model. Comparing with HAL-based query expansion, this experiment can show the additional benefit to extract IF from HAL space.

Local IF-based query expansion with pseudo-relevance feedback: In contrast to the previous method, this method constructs a local context (HAL) space by using the top 50 feedback documents in response to a query, and thereafter deriving a query model via IF computation from this local collection. The fifty documents were retrieved by the basic LM model. The top 60 information flows are used to expand query model. This number is also determined empirically.

The experimental results are measured using average precision (AvgPr) and recall, which are calculated on top 1000 retrieved documents.

In the experiments using language models, we use the Lemur toolkit[1]. In our models, several parameters have to be determined: λ in formulas (8) and (9), and the other smoothing parameters involved in different smoothing methods (e.g. Dirichlet prior, etc.). These parameters could be tuned automatically, for example, using EM algorithm [17]. However, in this paper, we determine these parameters empirically. The results that we report here are the best ones we obtained. The corresponding parameters are indicated in the summary of results.

5.2 Experimental results

Tables 3-5 show the experimental results on AP89 with queries 1-50, AP88-89 with queries 101-150 and 151-200 respectively. The percentages in the table are the relative changes with respect to the basic LM without query expansion. In general, the methods under comparison perform very similarly in the three cases.

Smoothing on document model: In our basic methods, two different smoothing methods are used on document model: Dirichlet and Two-stage smoothing. These methods have shown good results in other studies [17], and they are robust. In our experiments, the parameters have been tuned so that we can obtain the best effectiveness for these basic methods. One can observe that the effectiveness reported here is slightly higher than those reported in the previous studies on the same collections. For the other models, we do not change these parameters from the basic methods, but try to set other parameters, namely the mixture weight λ_{co} and λ_{IF}. So the other models are not tuned to their best.

Query expansion with HAL or co-occurrence information: When HAL relationships are used for query expansion, we can obtain improvements of around 3-4% for each of the smoothing methods. This improvement is similar to those reported in other studies that use co-occurrences to extend document model [4]. This comparable effectiveness improvement suggests that one can use co-occurrence information to expand either document model or query model, and both lead to similar effects. Although the query expansion is performed online (while document expansion is performed offline), as we limit the number of expansion terms, this will not require too much additional time for query evaluation.

Query expansion with global IF relationships: In contrast, when we use IF to expand queries (LM with IF), the effectiveness is greatly improved for each smoothing method. The effectiveness reported here are higher than those reported in other studies on the same test collections [2]. This experiment shows that IF combined with LM can indeed add interesting terms into queries, which cannot be added using raw co-occurrence relationships. The main reason is that the application of IF relationships are more constrained than that of pairwise relationships. When a new term is added by expansion, all (or most of) the query terms are used as the premise of IF relationship. In comparison with pairwise relationships, IF relationships allow us to avoid applying term relationships in inappropriate contexts. Therefore, less noisy terms are added during query expansion. In addition, in this experiment, as the queries are short, the IF relationships used to do query expansion usually include all the original query terms. Therefore, the selected expansion terms are chosen according to the whole query in a similar way to [13].

[1] The Lemur toolkit for language modeling and information retrieval: http://www.lemurproject.org/

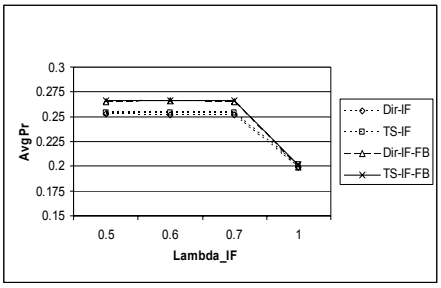

Figure 1. Effect of λ_{IF} on AP89, Q1-50

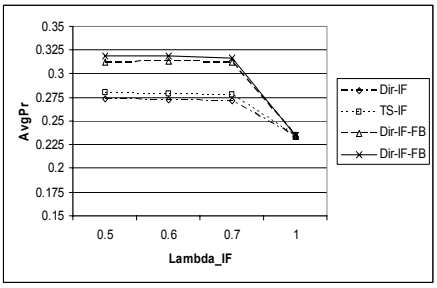

Figure 2. Effect of λ_{IF} on AP88-89, Q100-150

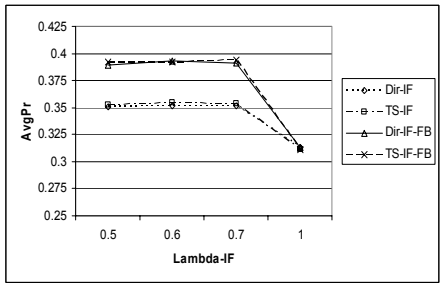

Figure 3. Effect of λ_{IF} on AP88-89, Q151-200

Table 3. Comparison between different models on AP89 collection for queries 1-50

		Basic LM	LM with HAL	LM with IF	LM with IF & FB
AvgPr	Dirichlet	(μ=2000) 0.1991	(λ_{co}=0.5) 0.2046 (++3%)	(λ_{IF}=0.5) 0.2524 (++27%)	(λ_{IF}=0.6) 0.2663 (++34%)
	Two-Stage	(λ=0.7) 0.2013	(λ_{co}=0.5) 0.2056 (++2%)	(λ_{IF}=0.6) 0.2539 (++26%)	(λ_{IF}=0.5) 0.2664 (++32%)
Recall	Dirichlet	1557/3301	1602/3301 (++3%)	2246/3301 (++44%)	2356/3301 (++51%)
	Two-Stage	1565/3301	1602/3301 (++2%)	2221/3301 (++42%)	2372/3301 (++52%)

Table 4. Comparison between different models on AP88-89 collection for queries 101-150

		Basic LM	LM with HAL	LM with IF	LM with IF & FB
AvgPr	Dirichlet	(μ=2000) 0.2338	(λ_{co}=0.5) 0.2435 (++4%)	(λ_{IF}=0.5) 0.2738 (++17%)	(λ_{IF}=0.6) 0.3130 (++34%)
	Two-Stage	(λ=0.7) 0.2347	(λ_{co}=0.5) 0.2451 (++4%)	(λ_{IF}=0.5) 0.2806 (++20%)	(λ_{IF}=0.6) 0.3185 (++36%)
Recall	Dirichlet	3160/4805	3258/4805 (++3%)	3717/4805 (++18%)	3893/4805 (++28%)
	Two-Stage	3130/4805	3220/4805 (++3%)	3729/4805 (++19%)	3900/4805 (++25%)

Table 5. Comparison between different models on AP88-89 collection for queries 151-200

		Basic LM	LM with HAL	LM with IF	LM with IF & FB
AvgPr	Dirichlet	(μ=1000) 0.3135	(λ_{co}=0.5) 0.3235 (++3%)	(λ_{IF}=0.7) 0.3516 (++12%)	(λ_{IF}=0.6) 0.3927 (++25%)
	Two-Stage	(λ=0.7) 0.3107	(λ_{co}=0.5) 0.3203 (++3%)	(λ_{IF}=0.6) 0.3540 (++14%)	(λ_{IF}=0.7) 0.3942 (++27%)
Recall	Dirichlet	3434/4933	3486/4933 (++2%)	3599/4933 (++5%)	3859/4933 (++12%)
	Two-Stage	3446/4933	3505/4933 (++2%)	3625/4933 (++5%)	3841/4933 (++11%)

Query expansion with local IF relationships: The last column (LM with IF & FB) shows the benefit of combining IF with pseudo-relevance feedback. This combination results in a set of local IF relationships extracted from the subset of documents that are closely related to the query. The local IF performs generally better than global IF. The comparison between expansion with global and local IF relationships is similar to that between global and local context analysis [16].

To see the difference between global and local IF relationships, we show below the 20 strongest terms in the expanded query from "space program" using local IF relationships:

space:0.987 program:0.759 nasa:0.758 shuttle:0.682 mission:0.644 launch:0.611 station:0.574 astronauts:0.568 earth:0.546 flight:0.543 new:0.533 satellite:0.516 president:0.506 national:0.496 billion:0.491 long:0.490 orbit:0.489 manned:0.488 bush:0.485 agency:0.478

We can see in this example that the terms with strong degrees are more relevant to the query than in the previous example (shown in section 4.3) with global IF relationships.

We can also compare this experiment with the previous studies on query expansion using feedback documents [8, 9, 18]. Our method shows higher effectiveness on the same test collections (AP88-89, queries101-150). This confirms the advantage to extract explicit term relationships from feedback documents, instead of using them as term distributions.

5.3 Effect of smoothing with term relationships

In order to see the impact of expanding query model with a relation model, we change the value of the smoothing factor λ_{IF} in the series of experiments with IF relationships. The results are shown in Figures 1-3.

As we can see, in all the cases, when λ_{IF}<1 (i.e. when the relation model is combined to some extent), we see clear improvements in the retrieval effectiveness compared to the case of λ_{IF}=1 (i.e. the basic model without relation model). In addition, we also see very steady effectiveness for λ_{IF} in the range of 0.5-0.7. These results suggest that smoothing by the relation model of IF is a useful and robust approach.

6. CONCLUSIONS

Language modeling emerges as an appropriate and effective framework for IR. However, most of the models assume term independence. As a consequence, they ignore relationships between terms (e.g. synonymy) which may enhance retrieval performance.

In some recent models, pseudo-relevance feedback has been used to create a better document or query model. However, no explicit term relationships are extracted, and the feedback documents only serve as different term distributions.

On the other hand, in a more classical setting, explicit term relationships have been used in query expansion in order to obtain a better expression of the query (or information need). However, the same approach has not been implemented in a LM setting.

In this paper, we propose an approach to LM which integrates the idea of query expansion. Term relationships are used to derive a new query model. Two specific types of term relationship are considered in this paper: co-occurrence relationships (or HAL relationships) and inferential relationships derived from information flow. We show that the idea of query expansion with term relationships can be naturally implemented in LM. Our experiments on TREC test collection show that such a query expansion is beneficial to IR. In addition, when IF relationships are used, query expansion is carried out in a context-dependent manner. This allows us to make a better selection of the expansion terms appropriate to the given query, and provides an effective way to deal with term ambiguity.

As IF relationships are extracted from documents, the approach can also be combined with the idea of pseudo-relevance feedback. Our experiments show that such a combination allows us to extract query-centered IF relationships. These relationships turn out to be better than the IF relationships computed from the whole document collection.

The present study shows the feasibility of integrating query expansion in LM. Several aspects can be improved: (1) In our current experiments, the parameters are set empirically. In fact, they can be tuned automatically using a mechanism, such as EM. (2) In our model, a query is still discomposed into single words (or unigram). This means that we have not considered the possible links between words in the query. This makes arise the question about query representation by words. It is often believed that words are not the best representation units. A possible solution would be to combine our approach with that of [6], so that term relationships are considered both between and within document and query. (3) Finally, we will test our approach on more test collections.

ACKNOWLEDGMENT

The work reported in this paper has been funded in part by an NSERC CRD grant (Canada), the Research Exchange Program of University of Montreal, and the Distributed Systems Technology Center (DSTC).

REFERENCES

[1] A. Berger and J. Lafferty (1999). Information Retrieval as Statistical Translation. *In Proceedings of the 22th ACM SIGIR Conference on Research and Development in IR,* pp.222-229.

[2] P. D. Bruza and D. Song (2002). Inferring Query Models by Computing Information Flow. *In Proceedings of the 11th International ACM Conference on Information and Knowledge Management,* pp.260-269.

[3] C. Burgess, K. Livesay and K. Lund (1998). Explorations in Context Space: Words, Sentences, Discourse. *Discourse Processes,* 25(2&3), 211-257.

[4] G. Cao, J. Y. Nie and J. Bai (2005). Integrating Term Relationships into Language Models. *In Proceedings of the 28th ACM SIGIR Conference on Research and Development in IR*, pp.298-305.

[5] W.B. Croft and J. Lafferty (2002). Language Models for Information Retrieval. *Kluwer Int. Series on Information Retrieval, Vol. 13*, Kluwer Academic Publishers.

[6] J. F. Gao, J. Y. Nie, G. Wu and G. Cao (2004). Dependence Language Model for Information Retrieval. *In Proceedings of the 27th ACM SIGIR Conference on Research and Development in IR*, pp.170-177.

[7] J. F. Gao, J. Y. Nie, J. Zhang, E. Xun, M. Zhou and C. Huang (2001). Improving Query Translation for CLIR using Statistical Models. *In Proceedings of the 24th ACM SIGIR Conference on Research and Development in IR*, pp. 96-104.

[8] J. Lafferty and C. Zhai (2001). Document Language Models, Query Models, and Risk Minimization for Information Retrieval**.** *In Proceedings of the 24th ACM SIGIR Conference on Research and Development in IR*, pp.111-119.

[9] V. Lavrenko and W. B. Croft (2001). Relevance-based Language Models. *In Proceedings of the 24th ACM SIGIR Conference on Research and Development in IR*, pp.120-127.

[10] K. Lund and C. Burgess (1996). Producing High-dimensional Semantic Spaces from Lexical Co-occurrence. *Behavior Research Methods, Instruments, & Computers,* 28(2), 203-208.

[11] K. Ng. (1999). A Maximum Likelihood Ratio Information Retrieval Model. *In TREC-8 Workshop notebook*.

[12] J. Ponte and W. B. Croft (1998). A Language Modeling Approach to Information Retrieval. *In Proceedings of the 21st ACM SIGIR Conference on Research and Development in IR*, pp.275-281.

[13] Y. Qiu and H. P. Frei (1993). Concept Based Query Expansion. *In Proceedings of the 16th ACM SIGIR Conference on Research and Development in IR*, pp.160-169.

[14] H. Schütze and J. O. Pedersen (1997). A Co-occurrence based Thesaurus and Two Applications to Information Retrieval. *Information Processing and Management,* 33(3), 307-318.

[15] D. Song and P. D. Bruza (2003). Towards Context-sensitive Information Inference. *Journal of the American Society for Information Science and Technology (JASIST)*, Vol. 54, 321-334.

[16] J. Xu and W. B. Croft (1996). Query Expansion Using Local and Global Document Analysis. *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in IR*, pp.4-11.

[17] C. Zhai and J. Lafferty (2001). A Study of Smoothing Methods for Language Models Applied to Ad hoc Information Retrieval. *In Proceedings of the 24th ACM SIGIR Conference on Research and Development in IR*, pp.334-342.

[18] C. Zhai and J. Lafferty (2001). Model-based Feedback in the Language Modeling Approach to Information Retrieval. *In Proceedings of the 10th International Conference on Information and Knowledge Management*, pp.403-410.