Using Markov Chains to Exploit Word Relationships in Information Retrieval

Guihong Cao, Jian-Yun Nie and Jing Bai
Dept. IRO, University of Montreal
C.P. 6128, succursale Centre-ville, Montreal, Quebec, H3C 3J7 Canada
{caogui, nie, baijing}@iro.umontreal.ca

Abstract

Document expansion and query expansion aim to add related terms into document and query representations in order to make them more complete. However, most previous studies are limited in two respects: They use either query expansion or document expansion, but not both; expansion has been limited to directly related words. In this paper, we propose a more general approach: both document and query representations are expanded, and the expansion process also exploits indirect term relationships. The whole process is implemented through Markov chains. Our experiments show that each of these extensions brings additional improvements.

1. INTRODUCTION

Statistical language modeling (LM) has been widely used in information retrieval (IR) in recent years (Berger and Lafferty, 1999; Lafferty and Zhai, 2001; Ponte and Croft, 1998; Zhai and Lafferty, 2001b). One typical approach is to construct two language models, one for the query (query model) and another for the document (document model). Then the document is ranked according to the negative KL divergence (Lafferty and Zhai, 2001) between the two models. Using this approach, it is obvious that the retrieval effectiveness strongly depends on the quality of the two models. Poor models of document and query will lead to low retrieval effectiveness.

Several attempts have been made to improve either document model or query model. Smoothing is the basic method used to improve document model: the document model is usually smoothed with the whole collection, for example, by the Jelinek-Mercer smoothing method (Zhai and Lafferty, 2001a). This smoothing can avoid the problem of zero-probability for the missing words in the document, thus allows such a document to still be retrievable for a query containing the missing word. To some extent, the addition of the new terms into the document model extends the latter to a more complete one. However, the blind smoothing with the collection can also be problematic: the added terms are not always related to the document. For example, this smoothing process may assign a larger probability to "market" than "natural disaster" to a document about "flood", since the former term is more common in the whole collection. This example shows that the terms added into a document by a traditional document smoothing method are not always related to the document, but the frequent terms occurring in the whole collection.

On the other hand, there is a series of studies aiming to improve query model by exploiting feedback documents, either to construct a relevance model (Lavrenko and Croft, 2001) or a better query model (Zhai and Lafferty, 2001).

Despite the improvements brought by these approaches, the traditional LM approaches still suffer from the underlying assumption of term independence, which implies that a term of a query is independent from a different term in a document, which is obviously not true. For example, if "computer" appears in a query and "programming" in a document, the two words are related; so should be the document and the query.

Several studies have been conducted to relax the independence assumption (Bai et al., 2005; Berger and Lafferty, 1999; Cao et al., 2005; Lafferty and Zhai, 2001): The relationships between query terms and document terms are used to relate a document to a query, even though they contain different (but related) terms. From a broader point of view, in exploiting term relationships to relate a document to a query, we are indeed making inferences based on term relationships.

In the previous studies, inference has been implemented in LM either as document expansion or query expansion. From a model-based point of view, document expansion (query expansion) aims to estimate a more exhaustive and precise document model (query model). However, using only one of them may limit the inference ability. We argue that this limitation is not necessary. By using both, the inference capability can be increased. Indeed, the problem of inference can be compared to a search problem in AI. Using either goal-driven or data-driven search, one can explore less inference steps than a two-directional search, with the same constraint of resources. The idea of making inference on both document and query is similar. In this paper, we propose a general model, which extends both document and query representations through inferences based on word relationships.

A second limitation of the previous studies is that only inference using direct term relationships is allowed. In this paper, we further extend inference by using indirect term relationships. This is implemented using multi-stage Markov Chains (MC) (Brin and Page, 1998; Toutanova et al., 2004).

Our experiments on TREC collections show that each of the above extensions will lead to consistent improvements in retrieval effectiveness, and several ones among them are statistically significant. This allows us to conclude that a higher inference capability in IR is beneficial.

This paper is organized as follows. The next section briefly describes the existing LM approaches applied to IR. In section 3, we describe the general model combining both document and query model expansion. In section 4, we provide the details on how to estimate the expanded document and query models based on multi-stage MC. In section 5, we present a series of experiments conducted on three TREC collections. Section 6 compares our work with some related ones. Finally, we summarize our work and suggest some future research avenues in section 7.

2. Previous Approaches to LM for IR

2.1 Basic LM

The basic idea behind LM for IR is to construct a language model for each document and a language model for the query, and to measure their correspondence according to KL-divergence between the two models (Lafferty and Zhai, 2001). More formally, the following score function is defined between a document *D* and a query *Q*:

$$\begin{aligned}&\mathrm{Score}(\mathrm{Q},\mathrm{D}) = \sum_{\mathrm{w_i}\in\mathrm{V}} \mathrm{P}(\mathrm{w_i}|\mathrm{Q})\log\frac{\mathrm{P}(\mathrm{w_i}|\mathrm{D})}{\mathrm{P}(\mathrm{w_i}|\mathrm{Q})}\\&= \sum_{\mathrm{w_i}\in\mathrm{V}} \mathrm{P}(\mathrm{w_i}|\mathrm{Q})\log\mathrm{P}(\mathrm{w_i}|\mathrm{D}) + \mathrm{C}(\mathrm{Q})\\&\propto \sum_{\mathrm{w_i}\in\mathrm{V}} \mathrm{P}(\mathrm{w_i}|\mathrm{Q})\log\mathrm{P}(\mathrm{w_i}|\mathrm{D}) \qquad (1)\end{aligned}$$

where w_i is a word belonging to the vocabulary *V*, *P(.|Q)* and *P(.|D)* are respectively the query model and the document model. $\mathrm{C}(\mathrm{Q})$ is a constant independent of *D*; so it can be omitted for document ranking. While the query model can be estimated by Maximum Likelihood Estimation (MLE), which is done in most traditional LM approaches, the document model has to be smoothed, usually with the collection model, in order to avoid zero probability for the missing words in a document (Zhai and Lafferty, 2001).

Once both document and query models are estimated, the subsequent matching process using formula (1) has often been limited to a direct comparison between them, without making any further inference. To see more clearly that the traditional approach does not make any inference, let us consider the following example. Suppose that a document on "airbus" does not contain (in its original description) the term "airplane". Then this term will be attributed a small probability during the smoothing with the collection model. As a consequence, a query asking for "airplane" will not have zero probability in this document. However, this does not mean that one has been able to infer "airplane" from "airbus". Other documents unrelated to "airplane" (for example, a document about fishing) have also been attributed similar probabilities for the same term, due to the same smoothing process. As a consequence, the ranking of this document in comparison with the others is not much affected due to the smoothing process. This example shows that smoothing on document is not an inference process.

Intuitively, in the above situation, one would be able to take advantage of the known relationship between "airbus" and "airplane" during the smoothing process. The known relationship would allow us to infer that a document about "airbus" is also related to "airplane". Therefore, even though "airplane" does not appear in the document, its probability should be high (much higher than an unrelated term).

Several approaches have been proposed to make the above inference in LM, either through document expansion or query expansion. We will review some of them below.

2.2 Pseudo-Relevance Feedback

It is known that queries submitted by users are usually not complete descriptions of the information needs. So an MLE for query model is insufficient. Pseudo-relevance feedback is a mechanism often used to improve it. Several approaches have been proposed: the feedback documents (top *N* retrieved documents) can be used to train a new language model which is then mixed with the original query model (Zhai and Lafferty, 2001b) or they can be used to derive a relevance model (Lavrenko and Croft, 2001). In the mixture model, a new feedback model $P(w|F)$ is estimated from feedback documents, and then mixed with the original query model as follows:

$$P_F(w|Q) = \lambda_F P_{ml}(w|Q) + (1 - \lambda_F)P(w|F) \quad (2)$$

where $P_{ml}(w|Q)$ is the MLE probability of *w* in *Q*. The feedback model is estimated by EM in (Zhai and Lafferty, 2001b) in such a way that the likelihood of the feedback documents with respect to the query model can be maximized.

The new query model now contains new terms that are selected from the feedback documents. The expanded model is supposed to be a better description of the user's information need. Due to the added terms, the documents that do not contain the original query terms, but the new terms extracted from the feedback documents, can still be retrieved. To limit the size of the query model, one has to limit the number of terms extracted from the feedback documents (for example, 80 strongest related terms).

Despite its positive effects, pseudo-relevance feedback strongly relies on the assumption that related terms co-occur often in the feedback documents. Therefore, pseudo-relevance feedback exploits implicitly the term relationships encoded by their co-occurrences in the feedback documents. Although many useful term relationships manifest as co-occurrences in the feedback documents, there may be other useful relationships missing from these documents. Therefore, a natural question is how we can extend query expansion beyond the co-occurrence relationships within feedback documents. Previous studies have exploited explicitly several types of relationship between terms in different ways. We review several ones in the following section.

2.3. Model Augmentation via Expansion

(Berger and Lafferty, 1999) proposes to use relationships $t(q_i|w)$ between two terms w and q_i to expand the document model as follows:

$$P_E(q_i|D) = \lambda P(q_i|D) + (1-\lambda)\sum_{w \in D} t(q_i|w)P(w|D) \quad (3)$$

where $P(q_i|D)$ is a classical (smoothed) unigram model. The probability $t(q_i|w)$ is estimated as the translation probability from a pseudo-parallel corpus. (Cao et al., 2005) further extends this method by integrating other types of term relationship, namely, co-occurrence relationships and lexical relationships from WordNet.

The above method tries to create a new document model $P_E(.|D)$ by integrating term relationships. It can be called a "document expansion" approach. A similar approach can also be used for query expansion. For example, (Bai et al., 2005) used co-occurrence relationships, as well as inference relationships induced by information flow (Song and Bruza, 2003), to expand the query model.

Despite the fact that the above models are able to infer new terms according to term relationships, inference has been limited to one step, i.e. only directly related terms are inferred and added during expansion. For example, if we know that "C++" is related to "programming" and "programming" to "computer", previous approaches only allow to extend a document about "C++" to "programming", but not to "computer". In fact, this limitation is unnecessary. We can exploit the indirect relationship between "C++" and "computer" in order to obtain higher inference capabilities. A natural extension is to allow for multi-step inference. Markov Chain (MC) is a suitable mechanism to implement multi-step inferences.

MC has been widely used in several previous studies (Brin and Page, 1998; Toutanova et al., 2004, Minkov et al., 2006). In LM framework, (Lafferty and Zhai, 2001) also uses MC for query expansion. In that paper, transitions between terms are made via documents: a transition from a term to some documents, then from these documents to another term. This method can naturally incorporate the effect of pseudo-relevance feedback, because the transition from a term to documents is indeed a retrieval process, and that from document to term is similar to query expansion via feedback documents. However, this particular way of estimating term relationships may suffer from the following limitation: it is unable to incorporate other types of term relationships (e.g. those in a thesaurus). In practice, many methods have been developed for extracting various term relationships from text collections, and there are also manually built thesauri that can provide term relationships.

Therefore, in this paper, we will propose a more general model that can integrate term relationships of different types.

The integration of term relationships in MC has also been studied in (Collins-Thompson and Callan, 2005). However, this model does not exploit fully the capability of MC, and many heuristics have been used. The experimental results only show marginal improvements over traditional approaches. In this paper, we will propose a more general and principled MC model, in which all the parameters will be estimated automatically. Therefore, our model can be easily adapted to other data set.

In the following section, we describe a general framework based on LM to combine both document and query expansions.

3. General Model Combining Document and Query Expansions

As we mentioned earlier, both query expansion and document expansion can be viewed as inference processes: document expansion tries to infer some possible and related query terms and add them into the document description, while query expansion makes inference in the

opposite direction. In most previous studies one has been limited to using one of them. Even if conceptually, single-direction inference is sufficient if it is performed completely and correctly, in practice, several factors may undermine the process: 1). Term relationships applied to either the document or the query are limited (by the resource or by the capability of an automatic tool to unveil the relationships). 2) Term relations can contain noise, and they are often ambiguous. Therefore, document expansion or query expansion has always been limited to the strongest terms. In so doing, one can limit the danger of spanning in all possible directions during expansion. However, this also limits the inference power and a possible connection between a document and a query can remain hidden.

At this point, one can draw an analogy with the search problem in AI (Russell and Norvig, 2003). One-direction search can be limited to some steps, and this can make a possible connection between data and goal unseen. In comparison, if search is conducted in both directions: from data to goal and from goal to data, the chance to connect a data to a possible goal is higher. In our case, we also limit the additional terms to a small number. This is comparable to the limitation on search steps in AI. As in search in AI, a possible connection between document and query can have a higher chance to be unveiled using a two-directional inference than a one-direction inference.

Therefore, by combining query expansion with document expansion, the above problems can be alleviated: On one hand, applying partial term relationships to both document and query can help creating a bridge between them more easily than if they are applied to one of them. On the other hand, the expansion on both elements can create a stronger bridge between the desired document and the query than those created in wrong direction.

From the model point of view, the ultimate benefit of using both document and query expansions lies in better models for them. It can be expected that better document and query models could lead to a more accurate comparison between document and query, thus higher retrieval effectiveness.

Therefore, we propose to use both document expansion and query expansion in our method.

Let $\widehat{P}(w_i|D)$ and $\widehat{P}(w_i|Q)$ be the expanded document model and query model. Then the documents are ranked by the following negative cross entropy:

$$Score(Q,D) = \sum_{w_i \in V} \widehat{P}(w_i|Q) \log \widehat{P}(w_i|D)$$

where w_i is a word of vocabulary V. In practice, query expansion should be limited to a relatively small number of terms because of retrieval efficiency. Let E be the set of expansion terms selected, (e.g. 80 strongly related terms to Q), then the above equation can be simplified as:

$$Score(Q,D) = \sum_{w_i \in E \cup Q} \widehat{P}(w_i|Q) \log \widehat{P}(w_i|D) \quad (4)$$

Now the remaining problem is to estimate the expanded document and query models. In this paper, we aim to create better document and query models that fully exploit the available term relationships. As we mentioned in section 2, the existing methods usually have been limited to one step expansion. In the following section, we will propose models based on MC, which is capable of doing multi-step expansion and provides higher inference capability.

4. Query and Document Expansion Model using Markov Chain (MC)

Hereafter, we use upper-case letters to represent random variables and lower-case letters for constants. For example, W represents an arbitrary term while w represents a specified word.

As document expansion and query expansion are similar, in the following descriptions, we will mainly describe query expansion. Some differences for document expansion will be presented.

4.1 MC Model for Query Expansion

Markov Chain (MC) is a stochastic process having Markov property (Brémaud, 1999). Basically, a MC is defined by two probabilities: the initial probability to select a state, and the transition probability from one state to another. The final probability of a state is determined according to them.

4.1.1 General Formulation

At first glance, MC may be seen to be unrelated to query formulation or expansion. In fact, it is. We can well describe the user's query formulation process in a way similar to a MC.

A good query can be viewed as a good summary of an information need. For a specific information need, in order to create such a summary, the user has first to select a concept to describe; then a term to describe it. Once the first concept is described, he/she can select another related term to describe the same concept further; or choose the next concept to describe. This process corresponds exactly to a process of Markov Chain. This shows that MC is coherent with the query generation process. Let us now use the same process to simulate the formulation of a good query expression. We assume that the initial query Q is an approximation of the user's information need. We define a MC, M, on the set E of expansion terms to generate query terms. M has an initial distribution $P_0(w|Q)$ and a state transition probability $P(w_i|w_j,Q)$. Therefore, the generation of a query can be modeled by an MC as follows:

Step 0: The user chooses an initial word according to an initial distribution with respect to his/her information need. This can be approximated by $P_0(W|Q)$.

Step t: Given the word w_j selected at step t-1, the user chooses to add a word w_i. This selection can be made in two ways: the user can choose w_i related to an existing word w_j at probability $(1-\gamma)$, or to add it as a new unrelated word (i.e., reset to step 0) at probability γ. The selection of the related word is determined by the transition probability $P(w_i|w_j,Q)$. So the probability of selecting w_i according to both cases is:

$$\widehat{P}(w_i|w_j,Q) = \gamma P_0(w_i|Q) + (1-\gamma)P(w_i|w_j,Q) \quad (5)$$

This is the global transition probability to w_i at step t.

Note that in the above process, we assumed that words are generated from the initial model independently, which is a strong assumption. However, this assumption is generally adopted in LM approaches for the sake of feasibility.

At a higher level, the above process creates in fact another MC with the initial distribution $P_0(w_i|Q)$ and state transition probability $\widehat{P}(w_i|w_j,Q)$. We denote this higher-level MC by $\widehat{M}$. We allow the above transition process continue until reaching a fixed point. With this definition, $\widehat{M}$ is guaranteed to have a stationary distribution $\pi(w|Q)$ (Brémaud, 1999), which is expressed as follows:

$$\pi(w|Q) = \lim_{T\to+\infty} \widehat{P}_T(W=w|Q) = \gamma\sum_{t=0}^{\infty}(1-\gamma)^t P_t(W=w|Q) \quad (6)$$

where $P_t(W=w|Q)$ is the state of M after t-th update. The above process can also be interpreted as a random walk: The random walk starts from W_0 which is sampled according to the initial state probability $P_0(w|Q)$. At each step, it stops walking with a probability γ, or continues walking with probability $1-\gamma$. In the second case, it transits to another state according to the transition probability $P(w_i|w_j,Q)$.

According to its definition, the stationary distribution $\pi(w|Q)$ does not change with the step variable T. This distribution is considered to be the best statistical model that we can construct from the information available (i.e. Q and terms relations). In fact, a change of probability (by the user) can be interpreted as a piece of evidence that the current probability distribution is not

yet a good one, and the user wishes to modify it. For example, the user may have attributed too high a probability to a term, which turns out to be a poor descriptor. So the user wishes to reduce its importance. With the stationary probability distribution, no change is required anymore. So it corresponds to a query model with which the user is satisfied. Therefore, the stationary probability distribution simulates the situation in which the user is satisfied with the description of the information need. Thus, we define $\widehat{P}(w|Q)$ - the final query model, as $\pi(w|Q)$.

4.1.2 Parameters for query expansion

$\widehat{M}$ is uniquely determined provided that its initial distribution and transition probabilities are given (Brémaud, 1999). The final probability distribution can be derived from them with an iterative updating process as described before. Moreover, $\widehat{M}$ is derived from *M* (equation 5). Therefore, we only need to explain parameter estimation of *M*.

Initial Distribution

The initial distribution can be the maximum likelihood estimation model, i.e. $P_0(w|Q) = P_{ml}(w|Q)$. However, as a query is usually very short, it cannot depict the user's information need precisely. Therefore, we incorporate pseudo-relevance feedback to create a better initial query expression. The generation of a query term is now made from two sources: the original query and feedback documents. Let *F* be the set of top *N* feedback documents of query *Q*. Then the initial state distribution can be estimated as in the mixture model (Zhai and Lafferty, 2001b):

$$P_0(w|Q) = \lambda P_{ml}(w|Q) + (1-\lambda)P(w|F) \quad (7)$$

where $P(w|F)$ is the probability of *w* in *F* and λ is the coefficient of original query model (set to be 0.5) This feedback model can be estimated with EM algorithm (Dempster et al., 1977) by maximizing the likelihood of feedback documents given the query model, as in (Zhai and Lafferty, 2001b)**.**

Transition Probability

To estimate the transition probability $P(w_i|w_j, Q)$, a first approach is to assume that the transition from a word to another is independent from the query *Q*, and only depends on the relationship between the two words. Let us use $P_R(w_i|w_j)$ to denote the relationship between two words. Then we have $P(w_i|w_j, Q) = P_R(w_i|w_j)$.

Various methods exist for the estimation of $P_R(w_i|w_j)$. Here, we use the method proposed in (Cao et al., 2005), in which different types of relation are considered in the estimation of $P_R(w_i|w_j)$: co-occurrence relation and relations in WordNet. Let us describe this method briefly. Co-occurrence relationship $P_{CO}(w_i|w_j)$ is estimated according to the frequency of co-occurrence of two terms. It is defined as follows:

$$P_{CO}(w_i|w_j) = \frac{\max(c(w_i,w_j|W)-\delta,0)}{\sum_{w'} c(w',w_j|W)} + \frac{c(*,w_j|W)\delta}{\sum_{w'} c(w',w_j|W)} P_{add-one}(w_i|W)$$

$$P_{add-one}(w_i|W) = \frac{\sum_{j=1}^{|V|} c(w_i,w_j|W)+1}{\sum_{i=1}^{|V|}(\sum_{j=1}^{|V|} c(w_i,w_j|W)+1)} \quad (8)$$

where δ is the discount factor (set to be 0.7 in our experiments) and $c(w_i, w_j|W)$ is the count of co-occurrence of w_i and w_j within a window of fixed size (8 words, which is determined empirically). $P_{WN}(w_i|w_j)$ is defined for two terms that are connected by a relation in WordNet. In order to attribute a probability to this relationship, co-occurrences of the two terms in texts are used. So the estimation is similar to Equation (8) but with the constraint that w_i and w_j are

also connected by a relation in WordNet, and that they appear in the same paragraph. Then the two types of relationship are combined via the following LM smoothing:

$$P_R(w_i|w_j) = \lambda_1 P_{CO}(w_i|w_j) + (1-\lambda_1) P_{WN}(w_i|w_j) \quad (9)$$

where λ_1 is the smoothing coefficient.

However, the above estimation of $P_R(w_i|w_j, F)$ is indeed query-independent, which is not reasonable. An alternative is to complement the above estimation by another relation model estimated from the feedback documents. Indeed, feedback documents *F* are more related to the query than the other documents. So the term relations estimated from the feedback documents are indeed query-dependent. It has been shown in (Bai et al., 2005) that such query-dependent term relations are better than query-independent ones for the purpose of query expansion. We can also expect that query-dependent transition probability estimation is better than a query-independent one. Let $P_R(w_i|w_j, F)$ be the term relationship extracted from the feedback documents. $P_R(w_i|w_j, F)$ can be estimated in the same way as $P_R(w_i|w_j)$ described earlier, except that it only uses the feedback documents instead of the whole collection. Then the final transition probability can be defined by combining both estimations as follows:

$$P'(w_i|w_j, Q) = \lambda_2 P_R(w_i|w_j, F) + (1-\lambda_2) P_R(w_i|w_j) \quad (10)$$

where λ_2 is a smoothing coefficient.

Several coefficients have been used to combine different models: the probability γ in Equation (5) to stop random walk and two λ_i (*i*=1, 2) for smoothing. As we will see in Section 5.5, the retrieval effectiveness is relatively insensitive to γ. So, here we fix the value of γ and tune the other parameters. Several strategies can be used to optimize parameters: generative methods to maximize the likelihood of queries (or relevant documents) (Cao et al., 2005; Zhai and Lafferty, 2001b) and discriminative methods to optimize the mean average precision (MAP) (Gao et al., 2005) on some training data. Here we try to optimize MAP. We follow the discriminative training method used in (Toutanova et al., 2004), which defines an objective function to be optimized from the coefficients. Due to the space limit, we will not describe the process in detail. Interested reader can refer to (Toutanova et al., 2004) for details. Finally, we use Simulated Annealing algorithm (Kirkpatrick et al., 1983) to maximize the objective function.

4.2 MC Model for Document Expansion

The document expansion model is similar to the query expansion model. The only differences are as follows:

- The initial probability distribution is determined by a smoothed document model
- The transition probability only relies on the whole collection.

For the initial probability distribution, we use the unigram model with the following absolute discounting smoothing [20]:

$$P_0(w_i|D) = \frac{\max(c(w_i;D)-\delta, 0)}{|D|} + \frac{\delta |D|_u}{|D|} P_{ml}(w_i|C) \quad (11)$$

where δ is the discount factor (which is empirically set to 0.7), $|D|$ is the length of the document, $|D|_u$ is the count of unique terms in the document, and $P_{ml}(w_i|C)$ is the maximum likelihood probability of the word in the collection *C*.

Unlike query expansion, we do not have feedback documents. So we assume the transition probability is independent of the document and determine it according to term relationships in the whole collection, i.e.:

$$P(w_i|w_j, D) = P_R(w_i|w_j)$$

The term relationship $P_R(w_i|w_j)$ is estimated in the same way as in the query expansion model (Equation 9).

An alternative is to exploit term relationships extracted from document clusters. Similarly to the utilization of feedback documents, by using the term relations extracted from the cluster to which the document belong, we could also estimate document-dependent transition probabilities. However, in this study, we do not exploit this possibility.As for query expansion, the stationary probability $\pi(w \mid D)$ is used as the final document model $\hat{P}(w|D)$, i.e.:

$$\hat{P}(w|D) = \pi(w|D) = \gamma \sum_{t=0}^{\infty} (1-\gamma)^t P_t(w|D) \quad (12)$$

and

$$P_t(w|D) = \sum_{w' \in V} P_R(w|w') P_{t-1}(w|D) \quad (13)$$

where *w* and *w'* are words in the vocabulary *V*; $P_t(w|D)$ is the document model after *t*-th update. Equation (12) converges very fast because $\gamma \in [0,1]$. We thus only iterate 4 times to calculate $\hat{P}(w|D)$.

5. Experiments

Table 1. Statistics of Data Set

Coll.	Description	Size (MB)	# Doc.	Vocab. Size	Avg Doc Len.	Query		Avg test Qry Len
						Testing	Training	
AP	*Associate Press* (1988-90), Disks 2&3	729	242,918	245,748	244	topics 51-100 (Title+Desc.)	topics 101-150 + 201-250 (Title+Desc.)	13
WSJ	*Wall Street Journal* (1990-92), Disk 2	242	74,520	121,944	264	As AP	As AP	As AP
SJM	*San Jose Mercury News* (1991), Disk 3	287	90,257	146,512	217	As AP	As AP	As AP

Several previous experiments have already shown that both query expansion and document expansion can improve the retrieval effectiveness. The goal of our experiments is twofold:
- We want to test if the utilization of multi-step expansion can further improve retrieval effectiveness over one-step expansion;
- We want to see if the general model that combines document expansion and query expansion performs better than each of them alone.

5.1 Experiment Setting

We used three TREC collections to evaluate our models: AP, WSJ and SJM. Table 1 shows the statistical information of the collections.
All English documents and queries were processed in a standard manner: terms were stemmed using the Porter stemmer and stopwords were removed. The document set comes from the TREC disks 2 and 3. The version of WordNet we used for experiments is 2.0. For each word in the vocabulary of dataset, we extract its synonym, hypernym and hyponym from WordNet and build a pool of related terms for it. The processing is done offline. When counting the co-occurrences of terms in WordNet model, the pool is used to determine whether there is a relation between two terms. As we do not consider explicitly compound terms, all the compound terms in WordNet are decomposed into their component words.
The effectiveness of IR is mainly measured by the standard non-interpolated average precision (AvgP). For each query, we retrieve 1000 documents. The total recall (Rec.) for all 50 queries is shown as a complementary metric. We also calculated the t-test for statistical significance and conducted query-by-query analysis.
We used Lemur3.0 (Ogilvie and Callan, 2001) as the basic retrieval tool, which is extended to support our experiments.

5.2. Multi-step expansion vs. One-step expansion

Since query expansion is as demonstrative as document expansion, we just show the results of query expansion.

In this experiment, we compare the performance of one-step query expansion with the Multi-step **MC-QE** model. The two models compared here have the same parameters, i.e., the initial distribution and transition probabilities. The only difference is the number of inference steps. These models incorporate the feedback documents: one step vs. multiple steps.

Figure 1 shows the results on all three collections. The one-step query expansion corresponds to the left-most points. We observe that when we increase the inference steps, the effectiveness on all the three collections is also improved. In particular, the improvements between 1 and 5 steps are the most important. These increases are directly attributed to the increased steps of inference. They show clearly that multi-step inference is superior to one-step inference.

In figure 1, we also see that **MC** converges in less than 20 steps for all the collections. Since MC converges very fast and there is a small number of states (80 terms), the query expansion can be very efficient. In our experiments, we observed that **MC** model took very little additional time (less than 1 second for each query).

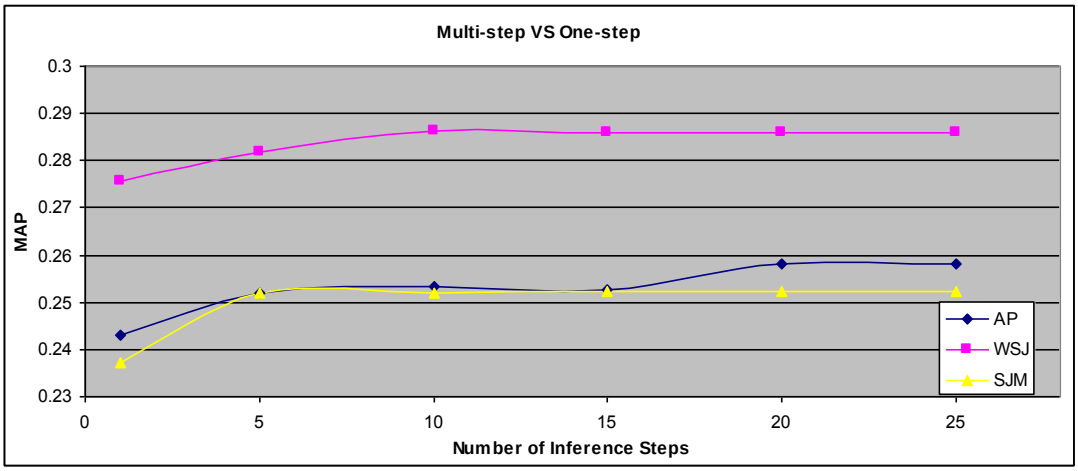

Figure 1: One-step QE Vs Multi-step MC-QE

5.3 Performance of the General Model

Table 2: Performance of General Model

Coll.		**UM**	**MC-DE**	%ch1	**MC-QE**	**GM**	%ch1	%ch2
AP	AvP.	0.1925	0.2138	+11.06**	0.2580	**0.2629**	+22.96**	+2.02
	Ret.	3289	3530		3994	**4064**		
WSJ	AvP.	0.2466	0.2590	+5.02*	0.2860	**0.2891**	+11.62**	+1.08
	Ret.	1659	1704		1794	**1845**		
SJM	AvP.	0.2045	0.2155	+5.37	0.2522	**0.2584**	+19.91**	+2.46
	Ret.	1417	1572		1621	**1742**		

Ret. is number of relevant documents retrieved; ch1 means "vs. UM"; ch2 means "vs. QE". * means the improvement is statistical significant (p-val <0.05) and **means very significant (p-val <0.001)

To investigate the performance of the general model described in section 3, we compared the general model with both document expansion and query expansion alone. Table 2 shows the results. Here **UM** is the unigram model, which does not perform any document/query expansion, and document model is smoothed using absolute discounting (Formula (11)). **MC-QE** and **MC-DE** are the query expansion and document expansion model based on MC described in section 4 respectively. **GM** is the general model combining both. From this table, we see that both **MC-DE** and **MC-QE** outperform **UM**. This shows that expanding either document or query will alleviate the mismatch between the query and the document to some degree. However the improvement scales of **MC-QE** are much higher than **MC-DE**. There are two possible reasons:

(1) The parameter estimation of **MC-DE** is coarser than **MC-QE**, in which we take the pseudo-relevance feedback documents into account. These documents are more informative than the whole collection. In fact, we use the feedback documents in two ways: to define the initial distribution and to define word relationships. This provides us with more related terms and defines the relationship more accurately. In contrast, to define the initial distribution of **MC-DE**, we only use the smoothed unigram model, which may assign fairly high probability of some common but irrelevant words. We leave the refinement of the initial distribution of **MC-DE** to our future work; (2) Document expansion is performed blindly according to the relationships extracted from the whole collection, while query expansion is performed according to query-oriented relationships. Therefore, less ambiguity exists in the second case.
We can also see that the combined model **GM** is better than using **MC-DE** or **MC-QE** separately. However, the improvement with this combination over **MC-QE** is not very large and not statistically significant. The smaller improvement scales over **MC-QE** is mainly due to the fact that **MC-QE** has already incorporated feedback documents that are highly helpful for the query model. The improvements over **MC-QE** are only due to the incorporation of **MC-DE**. This difference, even not very large, indicates that there is a potential advantage to combine both document expansion and query expansion.
Document expansion is a more complex operation than query expansion if we consider all the possible related terms, because there are more terms in a document than in a query. So the number of expansion terms can be very large. In our implementation, we limit the number of expansion terms to 80. In terms of time, query evaluation does not require much more additional time, because document expansion is performed offline. The additional time is due to the fact that each query term will correspond to more documents. Therefore, more time is required to produce the final ranking list.

5.4 The Impact of Different Query Expansion Methods

Table 3: Comparison different models for query expansion.

Coll.	**UM**		**QE**			**MixM**			**MC-QE**		
	AvP.	Rec.	AvP.	%chg1	Ret.	AvP.	%chg1	Ret	AvP.	%chg2	Ret.
AP	0.1925	3289/6101	0.1959	+1.76	3370	0.2350	+22.07 **	3700	**0.2580**	+9.79*	3994
WSJ	0.2466	1659/2172	0.2483	+0.68	1636	0.2731	+10.75 **	1730	**0.2860**	+4.72	1794
SJM	0.2045	1417/2322	0.2142	+4.74	1485	0.2298	+12.37 **	1526	**0.2522**	+9.75*	1621

Ret. is the number of relevant documents which are retrieved. chg1 means the improvement over UM and chg2 means the improvement over MixM. * means the improvement is statistical significant (p-val <0.05) and **means very significant (p-val <0.001)

The general model has two sub-models, expanded document and query model. Both of them are obtained based on MC. A lot of studies have been conducted on these topics. It is interesting to compare our expansion model with other existing models. Here we only examine the performance of query expansion. We compared the following models:
UM: unigram model. This is the basic LM without any expansion.

QE: the basic query expansion model, which only uses term relationships extracted from co-occurrences and from WordNet (for English). The feedback documents are not used. This experiments aims to show the contribution of inference in query expansion based solely on term relationships. In fact, the model can be expressed as follows:

$$P(w|Q) = \lambda P_{ml}(w|Q) + (1-\lambda)\sum_{q\in Q} P_R(w|q)P_{ml}(q|Q) \quad (14)$$

where $P_{ml}(w|Q)$ is the MLE probability of w in query *Q*, and λ is the coefficient which is set by manually trial. $P_R(w|q)$ is defined by Equation 9. We use 80 expansion terms in this experiment. We can also view QE as one-step inference model, in which the initial distribution is defined without considering the feedback documents.
MixM: query expansion with the mixture model (Zhai and Lafferty, 2001b). We used top 20 documents for feedback and chose 80 terms to add to the query.

MC-QE: as defined in section 5.3. All the common parameters of **MC-QE** with **MixM** are set to be the same. We also set γ in Equation 5 to be 0.3 for all three collections. All other parameters are tuned by discriminative training described in section 4.1.

Table 3 compares the three models. We can see that the **QE** only outperforms marginally the unigram model. This result is not really surprising and it is consistent with several studies on query expansion (Voorhees, 1994).

We can see in the column **MixM** that the utilization of a feedback model to mix with the original query model is very effective. All the improvements are statistically significant. This result is similar to that of (Zhai and Lafferty, 2001b).

What is interesting to observe is that term relationships used for query expansion (**MC-QE**) can further improve the effectiveness. The improvements over MixM are statistically significant in 2 test sets out of 3.

The additional power of **MC-QE** vs. **MixM** can also be seen as follows: in fact, **MixM** can be viewed as a special case of **MC-QE** with the number of random walk set to 0. Then the difference between **MC-QE** and **MixM** is directly attributed to the additional steps of expansion through the random walks, which tries to re-estimate term probabilities. During the process, **MC-QE** increases the probabilities of important terms and their related terms and decreases those that are wrongly attributed a high probability. At the end, we obtain a more accurate model. This experimental result confirms the advantage of multi-step inference for query expansion.

5.5 Sensitivity to stopping probability in Random Walk

Our model does not optimize the stopping probability (γ in Equation 5). In the experiments reported in the previous tables, γ is set at 0.3. We mentioned earlier that the retrieval effectiveness is not very sensitive to γ. Here we show some experimental evidence for it. In these experiments, we change γ from 0 to 1.0 and compare **MC-QE** with **MixM** for all the three collections. Figure 1 shows the results on English and Chinese collections. We observe that **MC-QE** outperforms **MixM** for all the values in the range $\gamma \in [0.1,1)$. Therefore, the performance of **MC-QE** is fairly good even though γ is not optimal.

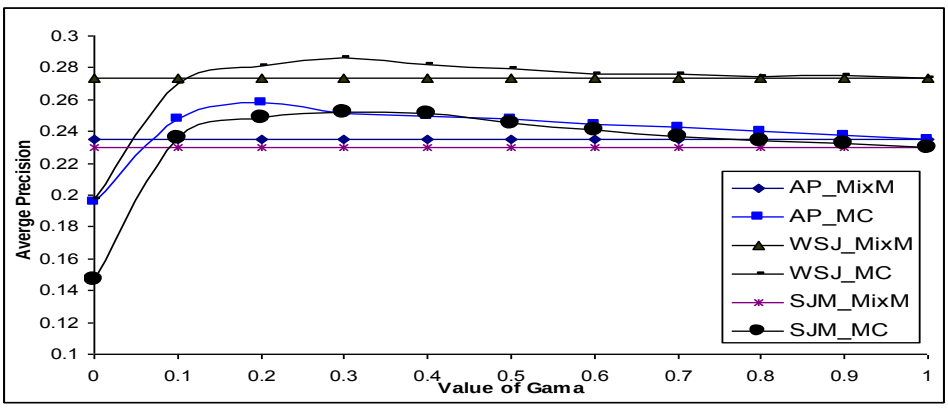

Figure 2: Sensitivity of γ for MC performance

6. Related work

Query expansion has been studied for a long time in IR (Sparck Jones, 1971). With classical IR models (e.g. vector space model), it produced variable results (Peat and Willett, 1991). Recently, several models for both document and query expansions have been proposed within the LM framework (Zhai and Lafferty, 2001b; Lavrenko and Croft, 2001; Cao et al., 2005; Bai et al., 2005). In particular, (Bai et al., 2005) exploited word relationships to expand the document and the query respectively. The experiments have produced encouraging results. In comparison with

the previous attempts, in this paper, we proposed to extend the expansion process further on the two following aspects:

- We used MC to enable multi-step inference during expansion in order to make more complete inference
- We proposed a general model to perform both document expansion and query expansion simultaneously

Each of the above extensions has resulted in some improvements in retrieval effectiveness.
MC has also been used in some previous studies in IR. For example, (Collins-Thompson and Callan, 2005) used it for query expansion. However, our method is different from theirs in several ways: first, we do not use many heuristics as in their work. We followed a more principled development and the parameters in the model are tuned automatically with a discriminative training method. Second, we incorporated pseudo-relevance feedback within the initial distribution of MC, which turns out to be essential to the retrieval effectiveness, while in their work, it is defined heuristically. Third, their work does not define a query model explicitly, while we did. So our approach based on MC has been much extended from their work.

7. Conclusions and Future Work

Many previous studies on LM assumed independence between terms. This assumption has two consequences: the terms within the same query or documents are assumed to be independent, and the terms in a document are assumed to be independent from different terms in a query. Both assumptions are not true in reality. In this paper, we proposed an approach to document and query expansion, which considers term relationships in both cases: A document term can be related to a different query term by applying term relationships; and a related term derived from a query term also depends on the other terms in the query (this is implied by our query-dependent transition probability).
Document and query expansions have been investigated separately in several previous studies. The models we proposed in this paper further extend these studies in several ways:

- Document and query expansion is not limited to one step as in other studies, but can perform multiple steps;
- Document and query expansions are performed simultaneously.

All the above extensions integrate additional inference capabilities into the IR model, allowing us to retrieve documents described with different but related terms.
Our experiments have examined each of the above aspects. It turns out that each addition brought some improvements to the retrieval effectiveness. These results show that we can create better document and query models by performing full inferences using term relations, and that it can be beneficial to improve both document and query models simultaneously.
The proposed models can be further improved to integrate more inference capabilities. For example, the same term relationships are used regardless to the area of the query. Although relevance feedback allowed us to restrict the expansion within the area of the query, a possible further improvement is to try to determine related terms to the whole query instead of to query terms as in (Bai et al. 2006). This means to extract more complex and context-dependent term relationships such as (Java, computer)→programming, instead of being limited to those between a pair of words such as Java→programming or Java→coffee. We will investigate this problem in the future.

Reference:

Bai, J., Song, D., Bruza, P., Nie, J.-Y. and Cao, G. (2005). Query Expansion Using Term Relationships in Language Models for Information Retrieval. In *Proceedings of the 14th CIKM*, pp. 688-695.

Bai, J., Nie, J.Y., Cao, G. (2006). Context-dependent term relations for information retrieval, In *Proceedings of EMNLP*, pp. 551-559.

Berger, A. and Lafferty, J. (1999). Information retrieval as statistical translation. In *Proceedings of the 1999 SIGIR*, pages 222-229.

Brémaud, P. (1999) Markov chains: Gibbs fields, Monte Carlo Simulations, and Queues. *Springer-Verlag.*

Brin, S., and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *WWW7/Computer Networks and ISDN Systems*, 20: 107-117.

Cao, G., Nie, J.-Y., and Bai, J. (2005). Integrating word relationships into language modeling. In *Proceedings of the 2005 SIGIR*, pp. 298-305

Collins-Thompson, K, and Callan, J. (2005). Query Expansion Using Random Walk Models. In *Proceedings Proceedings of the 14th CIKM*, pp.704-711

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1-38.

Foo, S. and Li, H. Chinese word segmentation and its effect on information retrieval. *Information Processing and Management*, 41(1), pp. 161-190.

Gao, J., Nie, J.-Y., Wu, G., and Cao, G. (2004). Dependence Language Model for Information Retrieval. In *Proceedings of the 2004 SIGIR* , pp. 170-177.

Gao, J., Qi, H., Xia, X., and Nie, J.-Y. (2005). Linear discriminative model for information retrieval. In *Proceedings of the 2005 SIGIR*, pp. 290-297

Kirkpatrick, S., Gelatt C., and Vecchi M., 1983. Optimization by Simulated Annealing. *Science*, 220(4598): 671-680.

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 2001 SIGIR*, pp. 111-119.

Lavrenko, V. and Croft, W.B. Relevance-Based Language Models. In *Proceedings of the 2001 SIGIR*, pp. 120-127.

Miller, D., Leek, T. and Schwartz, R. (1999). A hidden Markov model information retrieval system. In *Proceedings of the 1999 SIGIR*, pp. 214-222.

Minkov, E., Cohen, W., and Ng, A. (2006). Contextual Search and Name Disambiguation in E-mail Using Graph, In *the Proceedings of SIGIR 2006*, pp. 27-34.

Nie, J.-Y. (1988). An Outline of a General Model for Information Retrieval Systems. In *Proceedings of the 1988 SIGIR*, pp. 495-506

Ogilvie, P. and Callan, J. (2001). Experiments using the lemur toolkit. In *the Proceedings of TREC-10*, pp.103–108.

Peat, H. J. and Willett, P. 1991. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5). pp. 378-383

Ponte, J. and Croft, W.B. (1998). A language modeling approach to information retrieval. In *the Proceedings of the 1998 SIGIR*, pp. 275-281.

Russell, S., Norvig, P. (2003). Artificial Intelligence: A Modern Approach. 2nd Edition, Prentice Hall.

Song, D. and Bruza, P. (2003). Towards Context-sensitive Information Inference. *Journal of the American Society of Information Science and Technology*, 54: 321-334.

Sparck Jones, K. 1971. Automatic Keyword Classification for Information Retrieval. Butterworths.

Toutanova, K., Manning, C. and Ng, A. (2004). Learning Random Walk Models for Inducing Word Dependency Distributions. In *the Proceedings of the 21st International Machine Learning Conference, ACM Press, 2004*

Voorhees, E. (1994) Query Expansion Using Lexical-Semantic Relations, In *the Proceedings of the 1994 SIGIR,* pp. 61-69.

Zhai, C, and Lafferty, J. (2001a). A Study of Smoothing Methods for Language Models Applied to Information Retrieval. In *the Proceedings of the 2001 SIGIR,* pp. 334-342.

Zhai, C. and Lafferty, J. (2001b). Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *the Proceedings of the 10th CIKM,* pp. 403-410.