Dependence Language Model for Information Retrieval[1]

Jianfeng Gao[*], Jian-Yun Nie[**], Guangyuan Wu[#], Guihong Cao[#]

[*] Microsoft Research, Asia, Email: jfgao@microsoft.com
[**]Université de Montréal, Email: nie@iro.umontreal.ca
[#]Tianjin University, China.

Abstract

This paper presents a new dependence language modeling approach to information retrieval. The approach extends the basic language modeling approach based on unigram by relaxing the independence assumption. We integrate the linkage of a query as a hidden variable, which expresses the term dependencies within the query as an acyclic, planar, undirected graph. We then assume that a query is generated from a document in two stages: the linkage is generated first, and then each term is generated in turn depending on other related terms according to the linkage. We also present a smoothing method for model parameter estimation and an approach to learning the linkage of a sentence in an unsupervised manner. The new approach is compared to the classical probabilistic retrieval model and the previously proposed language models with and without taking into account term dependencies. Results show that our model achieves substantial and significant improvements on TREC collections.

Categories and Subject Descriptors

H [**3**]: 3 – *Retrieval models*

General Terms

Algorithms, Measurement, Theory

Keywords

Language Model, Dependence, Parser, Information Retrieval

1. Introduction

The *independence assumption* is one of the assumptions widely adopted in probabilistic retrieval theory. It states that terms are statistically independent from each other. Although this assumption makes the development of retrieval models easier and the retrieval operation tractable, it does not hold in textual data. This issue has motivated much research on term dependencies over the last decades. Most dependence models, however, have not delivered consistent improvements in effectiveness in large scale retrieval experiments. There are two reasons for this. First, it is practically difficult to estimate dependencies on a large scale. Second, it is theoretically challenging to integrate both single words and dependencies in a weighting schema. Without a theoretically motivated integration model, documents containing dependencies (e.g. phrases) may be over-scored if they are weighted in the same way as single words [18].

Some language modeling approaches also try to incorporate word dependency by using bigrams (e.g. [27]). However, while some of the word dependencies exist between adjacent words, others are more distant. A bigram model can hardly cover these latter dependencies. On the other hand, many "noisy" dependencies (those that are not truly connected) will also be assumed between adjacent words in a bigram model. At the end, the bigram language model showed only marginally better effectiveness than the unigram model.

Instead of assuming a dependency between every pair of adjacent words, we believe that on the one hand, more distant dependencies should be taken into account; and on the other hand, only the strongest dependencies should be recognized in order to make the approach tractable.

This paper presents a new method of capturing word dependencies. We extend the state-of-the-art language modeling approaches to information retrieval [21, 24, 33] by introducing a dependency structure, called *linkage*, which is inspired by the link grammar [8, 20]. The linkage structure assumes that term dependencies in a sentence form an acyclic, planar graph, where two related terms are linked. This dependency structure limits the dependencies to the most important relationships that are useful for retrieval. This not only reduces the processing time but also limits the estimation errors in computing dependence scores due to the sparse data problem. In our implementation, an existing dependency parsing [5] and some learning techniques (i.e. expectation maximization – EM) are extended to detect the linkage of a term sequence (which is not necessary a grammatical sentence) in an unsupervised manner. At the retrieval step, the linkage detected in a query is also expected to be generable from the linkage of a relevant document. Therefore, our model incorporates the requirements not only on words as in a unigram model, but also on linkage between words. In comparison with the bigram model, we will show that our model is a generalization of this latter.

In the rest of this paper: Section 2 reviews previous research trying to relax the independence assumption. Section 3 presents our dependence model and introduces several modeling assumptions to make the model tractable for information retrieval tasks. Section 4 presents in detail the methods of model parameter estimation, including the smoothing method and an approach to unsupervised learning of the linkage. A series of experiments on TREC collec-

SIGIR'04, July 25-29, 2004, Sheffield South Yorkshire, UK.
Copyright 2004, ACM 1-58113-881-4/04/0007…$5.00.

[1] This work was done while Wu and Cao were visiting Microsoft Research Asia. We thank Stephen Robertson and Chengxiang Zhai for their comments on early drafts of the paper. Thanks also go to the four anonymous reviewers, one of whom alerted us to the closely related work of Nallapati and Allan [23].

tions is presented in Section 5. The comparison of our approach to both the probabilistic retrieval models and the previous language models will show that our model achieves substantial and significant improvements. Conclusions and the contributions of this work are summarized in Section 6.

2. Previous Work

There has been a large amount of work dealing with term dependencies in both the probabilistic IR framework and the language modeling framework.

In classical probabilistic IR models, such as the binary independence retrieval (BIR) model [18], both queries and documents are represented as a set of terms that are assumed to be statistically independent. With respect to representations, two research directions can be taken in order to relax the independence assumption [9, 16].

The first is to produce a better model that integrates dependencies while using the same representation units (i.e. words). For example, there have been many attempts to improve the BIR model by considering various forms of term dependencies [7, 10, 17, 30]. In these approaches, documents and queries are still represented as a set of words/terms. Term dependencies are usually defined as statistical co-incidence between the terms on the scale of whole documents. Thus in principle, the formulation of relevance probabilities of documents given single terms should be elaborated to cover probabilities of document being relevant given combinations of terms [18]. Most dependence models, however, have not brought significant improvements in retrieval effectiveness. One important practical issue is that there would be too many term dependencies to be estimated. For example, a *blind* definition of term dependencies would be *every* possible combination of terms. As a consequence, the gain from improved independence assumption may not outweigh the loss from increased estimation errors. In our approach, we only retain the most probable term dependencies that are useful for information retrieval. These term dependencies are derived from the linkage on the scale of a sentence (or a query).

The second direction is to develop models for more detailed representations of queries and documents. In addition to terms which are single words or stems, *compound terms*, or phrases have also been used as representation units [1, 28]. Phrases are defined by collocations (adjacency, proximity) and selected on statistical ground (possibly with syntactic knowledge, such as POS tags of component terms). A phrase may then be treated as an indecomposable unit or a decomposable unit (i.e., both the phrase and its component single words are regarded as representation features). Unfortunately, the experiments do not provide a clear indication whether the retrieval effectiveness can be improved in this way. One possible reason is that, since phrases are different in nature from words, it may not be appropriate to apply the same weighting schemes for both of them. Otherwise, phrases are likely to be systematically over-scored in the independent model [18]. This problem is also referred to as the *weight normalization* problem.

Some tend to explain the unsuccessful trials of dependence models by the fact that dependency information might have already been (implicitly) captured in the classical probabilistic models [6]. It can also have been incorporated via the use of other techniques such as passage retrieval or query expansion as in Local Context Analysis [31]. So the need for introducing empirical dependency information is less important than had been generally thought. For example, Cooper [6] points out that in the case of the BIR model, the usual independence assumptions can be replaced by a weaker *linked dependence* assumption. This partially explains why the BIR model is so effective that most dependence models extended on top of it cannot bring much improvement.

Language modeling approaches to IR usually do not model relevance explicitly. Instead, documents are ranked according to their capability of generating the given query, i.e. $P(Q|D)$. The first language model proposed in [24] uses unigrams and does not consider any dependency between words. Since then, there have been several attempts to capture term dependencies [13, 22, 27, 29]. In most of the cases, unigrams are simply replaced by *bigrams* or *bi-terms*. In the latter approaches, two adjacent (ordered or no) words are assumed to be related. Therefore, these models have the capacity of representing some of the word dependencies. However, genuine dependencies do not only exist between adjacent words. They may also occur between more distant words such as between “distributed” and “network” in “distributed personal computer network”. Such distant dependencies cannot be correctly covered by a bigram or bi-term model. On the other hand, as we assume that every pair of adjacent words can be connected, a huge number of parameters have to be estimated. Because of data-sparseness, more errors will be generated. These errors may compromise the benefit that one may obtain from the term dependencies correctly detected.

This paper presents a more general dependence language model in which word dependencies are not restricted to adjacent words. However, given a sentence, we only consider the strongest word dependencies in order to reduce estimation errors. As we will show, our model is a generalization of the bigram model previously proposed.

3. A Dependence Language Model for IR

In the language modeling approach to information retrieval, a multinomial model over terms is estimated for each document D in the collection C to be searched. Then documents are ranked by the probability that a query $Q = (q_1,\ldots,q_m)$ would be observed as a sample from the respective document model, i.e. $P(Q|D)$. The unigram model estimates the query generation probability by $P(Q|D) = \prod_{i=1\ldots m}P(q_i|D)$. It assumes independence not only between two different query terms but also between multiple occurrences of the same query term [16].

3.1 A New Model

In our dependence modeling approach, we assume that term dependencies in a query (or a sentence) form a linkage: an acyclic, planar, undirected graph[2] where two (non stopword) related query terms are connected by a graph edge. An example of a query and its linkage is shown in Figure 1. Assuming the linkage L of query Q, the query generation is formulated as a two-stage process:

- The linkage L is firstly generated from the document according to the distribution $P(L|D)$;

[2] It is plausible to assume that the syntactic relation in most natural language sentences is acyclic and planar [5, 20]. The third property that the linkage is undirected will be discussed in this section, see also [32]. The linkage is also similar to the tree-based dependence in [17], where the term dependencies are incorporated into the classical retrieval model using a different method.

- The query Q is then generated according to the distribution $P(Q|L, D)$. In this second stage, the generation of a query term will also depend on the terms with which it is linked in L.

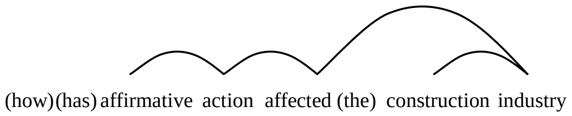
(how)(has) affirmative action affected (the) construction industry

Figure 1. A query example with its linkage, where stop words are bracketed.

The dependence language model in principle recovers the probability of the query $P(Q|D)$ over all possible linkages L. The basic dependence model is formulated as follows:

$$P(Q|D) = \sum_L P(Q,L|D) = \sum_L P(L|D)P(Q|L,D) \qquad (1)$$

We now introduce two assumptions to make the model of Equation (1) tractable.

First, following the common practice in statistical language modeling (e.g. [4]), we assume that the sum $\sum_L P(Q, L|D)$ over all the possible Ls is dominated by a single term L^* which is the most probable linkage of the query Q. Below we simply use L to represent L^*. Therefore, Equation (1) can be approximated as follows:

$$P(Q|D) = P(L|D)P(Q|L,D) \qquad (2)$$

such that $L = \arg\max_L P(L|Q)$.

In fact, with the above constraint on L, $P(L|D)$ can also be denoted as $P(L|Q, D)$. This assumption makes the model probabilistically illegitimate because $P(Q|D)$ in Equation (2) is no longer a true probability. However, the Probability Ranking Principle [18] suggests that any transformation of the probabilities, rather than the probability itself, can be used for document ranking provided that the transformation is order-preserving. As Equation (2) preserve the order of $P(Q|D)$ of Equation (1), we still write it as $P(Q|D)$ in Equation (2).

Second, we assume that each query term is dependent on exactly one related query term generated previously. Let L represent a set of related term pairs (q_i, q_j) where q_i is the governor of q_j. Since L is defined as an acyclic graph, each term q_j has exactly one governor q_i with the exception of the sentential head word q_h, which has no governor and governs the whole sentence, i.e., $q_j \neq q_h$. Therefore, $P(Q|L, D)$ can be decomposed as follows:

$$\begin{aligned}
P(Q|L,D) &= P(q_h|D)\prod_{(i,j)\in L} P(q_j|q_i,L,D) \\
&= P(q_h|D)\prod_{(i,j)\in L} \frac{P(q_i,q_j|L,D)}{P(q_i|L,D)} \\
&= P(q_h|D)\prod_{(i,j)\in L} \frac{P(q_i,q_j|L,D)P(q_j|L,D)}{P(q_i|L,D)P(q_j|L,D)}
\end{aligned}$$

We move the nominator term $P(q_j|L, D)$ outside the product operator, and assume $P(q|L, D) = P(q|D)$, i.e. the generation of a single term is independent of L. We then get:

$$\begin{aligned}
P(Q|L,D) &= P(q_h|D)\prod_{j\neq h} P(q_j|D)\prod_{(i,j)\in L} \frac{P(q_i,q_j|L,D)}{P(q_i|D)P(q_j|D)} \\
&= \prod_{i=1\ldots m} P(q_i|D)\prod_{(i,j)\in L} \frac{P(q_i,q_j|L,D)}{P(q_i|D)P(q_j|D)} \qquad (3)
\end{aligned}$$

We see that in Equation (3), q_h plays no special role: we would have arrived at the same result by starting from any term. This indicates that the direction of term dependencies does not matter in computing the query probability. Therefore, L can be represented as an undirected graph.

Finally, substituting Equation (3) into Equation (2) we have a new document scoring function. By taking the log, we can rewrite it in its final form:

$$\begin{aligned}
\log P(Q|D) = \log P(L|D) &+ \sum_{i=1\ldots m} \log P(q_i|D) \\
&+ \sum_{(i,j)\in L} MI(q_i,q_j|L,D) \qquad (4)
\end{aligned}$$

where

$$MI(q_i,q_j|L,D) = \log\frac{P(q_i,q_j|L,D)}{P(q_i|D)P(q_j|D)}.$$

3.2 Comparison with Other Models

We now discuss the similarities and differences between our model and the previously proposed dependence models. Our model not only covers most language model approaches as special cases, but also captures more useful information for document retrieval. For example, if we use independence assumption i.e., $L=\Phi$, only the second score term in the right-hand side of Equation (4) has a non-zero value. We then obtain the unigram language model. If we define a deterministic linkage L where each query term is dependent on its proceeding term, then $P(L|D)$ (or $P(L|Q,D)$) can be dropped for it always equals 1. In $MI(q_i,q_j|L,D)$ we also have $q_h = q_1$ and $i = j$-1. We therefore obtain the bigram language model similar to that described in [27]

$$P(Q|D) = P(q_1|D)\prod_{j=2\ldots m} P(q_j|q_{j-1},D)$$

Srikanth and Srihari [29] suggest that unlike language modeling for speech recognition [14], the language models for information retrieval need only to record co-occurrence of terms. Thus, they introduce *bi-term language models*. These models are similar to the bigram model except that the constraint of order in terms is relaxed. Therefore, a document containing *information retrieval* and a document containing *retrieval (of) information* will be assigned the same probability of generating the query. Some improvement using the bi-term model over the bigram model has been reported in [29].

However, most of the proposed extensions to the unigram model only consider dependencies between adjacent terms. Although some of the term dependencies can be captured, more distant dependencies are ignored. In our model, we do not impose a link between two adjacent words, but allow links between more distant words. These links are supposed to be the most important ones.

Most classical dependence models estimate term dependencies statistically on the scale of whole documents. Our model assumes a dependency structure (i.e., linkage) on the scale of a sentence by taking into account several linguistically motivated constraints (i.e., planar, acyclic). This would not only retain only those important term dependencies, but also allows us to apply efficient parsing techniques to detect term dependencies. The weight normalization problem described in Section 2 can also be resolved using our model in a systematic way. For instance, the first score term in

Equation (4) can also be viewed as a normalization factor that would penalize the scores of less probable term dependencies.

We recommend a comparison of our approach with that of [23]; the principal contrast lies in our method of estimating term dependencies in a sentence, and our use of the parsing score $P(L|D)$.

4. Parameter Estimation

In Equation (4), three terms have to be estimated: $P(L|D)$, $P(q_i|D)$ and $MI(q_i,q_j|L,D)$.

4.1 Estimating *P(L|D)*

Recall that L is the strongest linkage determined for Q. To determine a linkage for a query or for any sentence in a document, one may suggest using syntactic/semantic parsers designed for natural language processing. However, the difficulty arises because (1) the existing parsers often require syntactic and semantic knowledge of every word, and this is usually unavailable, especially for special words such as proper nouns; and (2) many queries are not grammatical sentences. Our solution is a statistical approach that incorporates some basic linguistic constraints on the form of the linkage, i.e. it is acyclic and planar. Below, we first present a parsing model, which assigns the probability of the linkage L given a query Q, $P(L|Q)$. Then we describe a statistical dependency parser which detects the most probable L in Q according to the parsing model: $L = \text{argmax}_L\ P(L|Q)$. We finally present how we create an annotated corpus with links for parsing model training in an unsupervised manner.

4.1.1 The Parsing Model

For the moment, we assume that there is an available training corpus where the linkage of each sentence is annotated. The creation of such corpus will be described in Section 4.1.3.

Our model is inspired by [5]. Let L be a set of probabilistic dependencies (or links) $l \in L$. Assuming that the dependencies are independent, we have the parsing model

$$P(L\,|\,Q) = \prod_{l \in L} P(l\,|\,Q) \qquad (5)$$

where $P(l|Q)$ is the dependency probability, estimated on the linkage-annotated training data. In particular, we count the relative frequency of link l between q_i and q_j given that they appear in the same sentence by:

$$F(R\,|\,q_i, q_j) = \frac{C(q_i, q_j, R)}{C(q_i, q_j)} \qquad (6)$$

where $C(q_i, q_j, R)$ is the number of times that q_i and q_j have a link in a sentence in training data, and $C(q_i, q_j)$ is the number of times that q_i and q_j are seen in the same sentence. The quantity of Equation (6) can be normalized to give the dependency probability in Equation (5). We however ignore the normalization factor because it would change neither the parsing results nor the ranking results in retrieval. That is, we assume $P(l|Q) \propto F(R|q_i,q_j)$. So the parsing model we used in our experiments is

$$L = \arg\max_L P(L\,|\,Q) = \arg\max_L \prod_{(i,j)\in L} F(R\,|\,q_i, q_j) \qquad (7)$$

Therefore, $P(L|D)$ (or $P(L|\text{Q}, D)$ more exactly) in Equation (2) can also be approximated by the quantity of this un-normalized parsing score because this approximation is order-preserving:

$$P(L\,|\,D) = P(L\,|\,Q, D) = \prod_{l \in L} P(l\,|\,D) \propto \prod_{(i,j)\in L} F(R\,|\,q_i, q_j)$$

In language modeling approaches to information retrieval, the parsing model (or more precisely $F(R|q_i, q_j)$ in our case) is usually estimated based on a single document. However, the maximal likelihood estimator in Equation (6) would be problematic because of data sparseness, i.e., $C(w_i, w_j, R)$ in a document may be too low to give a reliable estimate, or even worse it may be zero, leaving the estimate undefined. In our experiments, we smooth the estimate in two stages. First, we linearly interpolate the two parsing models, one of which is trained on the document D and the other on entire collection C:

$$F(R\,|\,q_i, q_j) = (1-\lambda)F_D(R\,|\,q_i, q_j) + \lambda F_C(R\,|\,q_i, q_j) \qquad (8)$$

where λ is the interpolation weight determined empirically. Second, for both $F_D(R|q_i,q_j,)$ and $F_C(R|q_i,q_j,)$ in Equation (8), we use the backoff schema similar to [5] for smoothing. The basic idea is to backoff the estimates based on less contextual information. Below, we ignore the notation difference between the estimates on the document E_D and those on the collection E_C, and only use the general form E, which will be estimated with respect to a document D or a corpus C. Let's define three estimates, E_1, E_{23}, and E_4

$$E_1(w_i, w_j) = \frac{\eta_1}{\delta_1} \quad E_{23}(w_i, w_j) = \frac{\eta_2 + \eta_3}{\delta_2 + \delta_3} \qquad E_4(w_i, w_j) = \frac{\eta_4}{\delta_4}, \qquad (9)$$

where

$\eta_1 = C(w_i, w_j, R)$, $\delta_1 = C(w_i, w_j)$,

$\eta_2 = C(w_i, *, R)$, $\delta_2 = C(w_i, *)$,

$\eta_3 = C(*, w_j, R)$, $\delta_3 = C(*, w_j)$, and

$\eta_4 = C(*, *, R)$, $\delta_4 = C(*, *)$

are counts in the appropriate context (i.e. either in a document or in a corpus). Here,* is a wild-card matching every word. The final estimate E (for both $F_D(R|q_i,q_j)$ and $F_C(R|q_i,q_j)$) takes the following form:

$$E = \lambda_1 E_1 + (1-\lambda_1)(\lambda_2 E_{23} + (1-\lambda_2)E_4) \qquad (10)$$

where λ_1 and λ_2 are smoothing parameters defined as

$$\lambda_1 = \frac{\delta_1}{\delta_1 + 1}, \text{ and } \lambda_2 = \frac{\delta_2 + \delta_3}{\delta_2 + \delta_3 + 1}.$$

4.1.2 The Parsing Algorithm

Given the parsing model, we use a standard bottom-up chart parsing algorithm to detect the most probable L given Q according to Equation (7). The following dynamic programming heuristics are used: if a link crossing or cycle is detected, the link with the lowest dependency probability in conflict is eliminated.

4.1.3 Unsupervised learning of L

This section describes how to create a training corpus annotated with *links* based on which the parsing model is estimated. Since there is no such corpus available for our purpose, we used an unsupervised learning method that discovers L of a given sentence.

Detailed description can be found in [12]. The principle is as follows.

We use a Viterbi iterative training procedure (an approximation of the EM training) for joint optimization of the parsing model and the linkage of training data. The method consists of three steps:

Step 1, initialize: We set a window of size *N* and assumed that each word pair within a headword *N*-gram constitutes an initial dependency. The optimal value of *N* is 3 in our experiments. That is, given a word trigram (w_1, w_2, w_3), there are 3 initial links: l_{12}, l_{13}, and l_{23}. From the initial links, we computed an initial dependency parsing model by Equations (5) and (6).

Step 2, (re-)parse the corpus: Given the parsing model, we used the Yuret's parser [32] to select the most probable linkage for each sentence in the training data. This parser successively eliminates the weakest conflicting link along with the parsing of a sentence. This results in an updated set of links.

Step 3, re-estimate the parameters of parsing model: We then re-estimated the parsing model parameters based on the updated link set. Steps (2) and (3) are iterated until the improvement in the probability of training data is less than a threshold.

Notice that Yuret's parser uses an approximation chart parsing algorithm. We use it for iterative training for its operating speed that is $O(n^2)$. The complexity of the abovementioned chart parser is $O(n^5)$. Although it cannot guarantee to find the optimal *L* for a given sentence, it has been demonstrated as a good approximation in practice [12, 32].

4.2 Estimating $P(q_i|D)$

We use the two-stage smoothing method proposed in [33] to estimate the unigram probability. First, the document language model is smoothed with a Dirichlet prior. Second, it is interpolated with a query background model, i.e., the model trained on the entire collection *C*. The final estimation is

$$P'(q_i \mid D) = (1-\lambda)P(q_i \mid D) + \lambda P(q_i \mid C)$$

$$= (1-\lambda)\frac{C_D(q_i)+C_C(q_i)}{\sum_{q_i} C_C(q_i)+\mu} + \lambda\frac{C_C(q_i)-\delta}{\sum_{q_i} C_C(q_i)} \qquad (11)$$

where μ is the parameter of the Dirichlet distribution, and δ is a constant discount – the mass that was stolen by the Good-Turning method, and was redistributed among the unigram probabilities of unseen terms in *C*. Good-Turing assumes that the number of unseen events is the same as the number of the events that occur once [15]. The final estimate to word unigram is

$P(q|C)= r^*/N$, where $r^*= (r+1)\, n_{r+1}/n_r$.

Here *r* is the number of times term *q* occurs in *C*, *N* is the total number of term occurrences in *C*, and n_r is the number of terms which occur *r* times in *C*.

4.3 Estimating $MI(q_i, q_j \mid L, D)$

Unlike the unigram probability, we do not use collection information in estimating term dependencies $MI(q_i, q_j|L, D)$. For all unseen links (q_i, q_j) in the document *D*, we simply assign $MI(q_i, q_j|L,D) = 0$, meaning that the two terms are independent in *D*, or in other words, knowing one term does not reduce the entropy of the other. The values of the seen term dependencies are estimated as

$$MI(q_i, q_j \mid L, D) = \log\frac{P(q_i, q_j \mid L, D)}{P(q_i \mid L, D)P(q_j \mid L, D)}$$

$$= \log\frac{C_D(q_i, q_j, R)/N}{(C_D(q_i, *, R)/N)(C_D(*, q_j, R)/N)}$$

$$= \log\frac{C_D(q_i, q_j, R)N}{C_D(q_i, *, R)C_D(*, q_j, R)} \qquad (12)$$

where $C_D(q_i, q_j, R)$ denotes the count of the link (q_i, q_j) in the document *D*, and $N= C_D(*,*,R)$.

5. Experiments

5.1 Experimental Setting

We evaluated the dependence language model approach described in the previous sections using six different TREC collections.[3] Some statistics are shown in Table 1. All documents have been processed in a standard manner: Terms were stemmed using the Porter stemmer and stop words were removed. The queries are TREC topics 202 to 250 (description field only)[4] on TREC disks 2 and 3. Those topics are "natural language" queries consisting of one sentence each of length 10 to 15 words. For different TREC collections, we remove those queries that have no relevant document.

Coll.	**Description**	**Size (MB)**	**# Doc.**	**# Query**
WSJ	*Wall Street Journal* (1990, 1991, 1992), Disk 2	248	74,520	45
PAT	U.S. Patents (1993), Disk 3	246	6,711	14
FR	*Federal Register* (1988), Disk 2	213	19,860	27
SJM	*San Jose Mercury News* (1991), Disk 3	291	90,257	48
AP	Associated Press (1988, 1989, 1990), Disks 2 and 3	484	158,240	49
ZIFF	Articles from *Computer Select disks*, Disks 2 and 3	532	217,940	32
ALL		2,014	567,528	49

Table 1. TREC collections.

The retrieval models in comparison, either language models or BIR models, contain free parameters that must be estimated empirically by trial and error. These parameters include smoothing parameters in language models and weights or constants in the BIR model. Therefore, we have applied an experimental paradigm called 2-fold cross validation. For each TREC collection, we divided the document set into two similar halves, e.g. even- and odd-numbered respectively, with one used for weight computation and the other for weight application. In our experiments, the retrieval results reported on each TREC collection (as shown in Tables 2 and 3) are combined by two sets of results on two halves of the collection, respectively. Each set of results on one half is obtained using the parameter settings optimized on the other half.

[3] It is desirable to use several small collections rather than one big collection, since it is known that retrieval performance varies a lot according to different collections.

[4] Topic 201 was not used for the TREC evaluations since it retrieved no relevant document (see for example [13]).

Models	**WSJ**			**PAT**			**FR**		
	AvgP	% change over BM	% change over UG	AvgP	%change over BM	% change over UG	AvgP	% change over BM	% change over UG
BM	22.30	--	--	26.34	--	--	15.96	--	--
UG	17.91	-19.69**	--	25.47	-3.30	--	14.26	-10.65	--
DM	**22.41**	+0.49	+25.13**	**30.74**	+16.70	+20.69	**17.82**	+11.65*	+24.96*
BG	21.46	-3.77	+19.82	29.36	+11.47	+15.27	15.65	-1.94	+9.75
BT1	21.67	-2.83	+20.99*	28.91	+9.76	+13.51	15.71	-1.57	+10.17
BT2	18.66	-16.32	+4.19	28.22	+7.14	+10.80	14.77	-7.46	+3.58

Table 2. Comparison results on **WSJ**, **PAT** and **FR** collections. * and ** indicate that the difference is statistically significant according to t-test (* indicates *p*-value < 0.05, ** indicates *p*-value < 0.02).

Models	**SJM**			**AP**			**ZIFF**		
	AvgP	% change over BM	% change over UG	AvgP	%change over BM	% change over UG	AvgP	% change over BM	% change over UG
BM	19.14	--	--	25.34	--	--	15.36	--	--
UG	20.68	+8.05	--	24.58	-3.00	--	16.47	+7.23	--
DM	**24.72**	+29.15*	+19.54**	25.87	+2.09	+5.25**	**18.18**	+18.36*	+10.38**
BG	24.60	+28.53*	+18.96**	**26.24**	+3.55	+6.75*	17.17	+11.78	+4.25
BT1	23.29	+21.68	+12.62**	25.90	+2.21	+5.37	17.66	+14.97	+7.23
BT2	21.62	+12.96	+4.55	25.43	+0.36	+3.46	16.34	+6.38	-0.79

Table 3. Comparison results on **SJM**, **AP** and **ZIFF** collections. * and ** indicate that the difference is statistically significant according to t-test (* indicates *p*-value < 0.05, ** indicates *p*-value < 0.02).

The performance of information retrieval is measured through the precision-recall pair. The main evaluation metric in this study is the non-interpolated average precision (AvgP). The significance tests (i.e. t-test) and query by query analysis are also employed.

5.2 Results

Tables 2 and 3 present our experimental results, where we compare our dependence model with probabilistic retrieval models including an implementation of the BIR model and state-of-the-art language modeling approaches with and without taking into account term dependencies.

BM represents the BIR model. We performed experiments using the Okapi system which is considered as a representative implementation of the BIR model. The retrieval approach models within-document frequencies by means of a mixture of two Poisson distributions [25]. It hypothesizes that occurrences of a term in a document have a stochastic element that reflects the distinction between those documents which are ‘about’ the concept (or ‘elite’) represented by the term and those which are not. For the great number of term weighting functions provided by Okapi, we choose BM2500 for it has achieved good performance in previous experiments [26].

UG is our implementation of the unigram language model approach to information retrieval proposed in [33]. It serves as the baseline language model approach in our experiments. Over all six TREC collections, UG achieves the performance similar to, or slightly worse than, that of BM. It has been observed that in general the classical probabilistic retrieval model and the unigram language model approach perform very similarly if both have been fine-tuned. The slightly worse performance of UG in our experiment might be due to our ‘over-tuned’ Okapi system (i.e. BM2500 have more weighting parameters to be tuned empirically).

DM is the dependence model described in Equations (2) to (4). We create the linkage annotated corpus in an unsupervised manner as described in Section 4.1, using all six TREC collections (i.e. the **ALL** collection in Table 1). We iterate the learning process two times. By comparing DM with BM and UG, we can see that our dependence model achieves substantial improvements in average precision in all six collections. In five out of six collections, the improvement of DM over UG is statistically significant i.e. *p*-value < 0.05 according to t-test. It indicates that the additional two terms in Equation (4), i.e., parsing score and term dependencies score, provide useful term dependency information for document retrieval. In our pilot study, we also compare the two versions of the dependence model of Equation (4) with and without the parsing score term. We find that the model with parsing score consistently outperforms the one without it. This may indicate the normalization capability of our dependence model described in Section 3: The parsing score serves as a normalization factor (or penalty) to balance the impact of single terms and term dependencies on information retrieval.

BG is our implementation of the bigram language modeling approach to information retrieval. The query generation probability is estimated by $P(Q|D) = P(q_1|D)\prod_{i=2\ldots m}P(q_i|q_{i-1}, D)$. It assumes that the query term is only depending on its one preceding term. To deal with the sparse data problem, we used two smoothing methods. First, we linearly interpolated the bigram models trained on the document *D* and the entire collection *C*, respectively. Second, for both bigram models, the bigram probability was linearly interpolated with the unigram probability. As described in Section 3, the bigram model is a special case of our dependence model by assuming a pre-defined linkage. The results show that this is a good assumption in practice: BG is slightly worse than DM in five out of six TREC collections but substantially outperforms UG in all collections. We then investigate in detail the linkages detected by the parser. It turns out that around 50% of the links are between two adjacent terms which are also captured by the bigram model. We can see here that a bigram model can only capture part of the interesting dependencies.

BT1 and **BT2** are our implementations of the bi-term language models originally described in [29]. They are approximations of the bigram language model by relaxing the constraint of term order. In BT1, the bi-term probability of the term pair (q_{i-1}, q_i), P_{BT1} is viewed as an average of bigram probability P_{BG} of the ordered pairs (q_{i-1}, q_i) and (q_i, q_{i-1}), where P_{BG} is the smoothed bigram probability given by the BG described above.

$$P_{BT1}(q_i \mid q_{i-1}, D) = \frac{1}{2}(P_{BG}(q_i \mid q_{i-1}, D) + P_{BG}(q_{i-1} \mid q_i D))$$

In BT2, the bi-term probability is computed as the ratio of the frequency of the term pair to the minimum of the frequencies of terms q_{i-1} and q_i.

$$P_{BT2}(q_i \mid q_{i-1}, D) = \frac{C_D(q_{i-1}, q_i) + C_D(q_i, q_{i-1})}{2 \times \min\{C_D(q_{i-1}), C_D(q_i)\}}$$

To handle the data sparse problem, the bi-term probability P_{BT2} is smoothed by unigram probability $P(q_i|D)$, which is in turn smoothed using its collection probability $P(q_i|C)$. As shown in Tables 2 and 3, though bi-term models outperform UG substantially, they do not outperform BG as presented in [29]. They also have a lower effectiveness than our model DM.

In summary, several conclusions can be drawn from the experiments.

- Our dependence model outperforms both the unigram language model and the classical probabilistic retrieval model substantially and significantly.
- In the language model approaches to information retrieval, models that capture term dependencies achieve substantial improvements over the unigram model.
- Bigram language model turns out to be a good approximation of the proposed dependence model in practice for it is simpler and achieves only slightly worse performance.
- Although bi-term language models are expected to be good approximations of the bigram language model, they have not delivered substantial improvements in effectiveness in our experiments.

5.3 Discussions on Term Dependencies with or without Linguistic Structure

Approaches to incorporating term dependencies in language modeling can be classified along the scale of how much linguistic structure being used. On one end of the scale, there are term co-occurrence models, which use no or very little linguistic information: Any two terms within a distance in the same document are assumed to have a dependency. These models have been proved not applicable in information retrieval as discussed in Section 2. In our study, we generate a term co-occurrence model (i.e. **CM** in Table 4) assuming that any term pair within a term trigram in a sentence has a link [11]. As shown in Table 4, although the size of CM (i.e., # of dependencies) is much larger, the improvement is very limited. On the other end of the spectrum, we have models that use sophisticated syntactic structure, such as dependency-based models [4, 5] and constituency-based models [2, 3]. They all use syntactic grammars for parsing and the parsing model is estimated from a manually annotated training data (i.e. UPennTree Bank). We have not evaluated these models in information retrieval due to their complexity. How to adopt them in information retrieval tasks is an open question. We leave this to our future work.

Our dependence model described in the previous sections falls between the two in the complexity of the linguistic structure it uses. In particular, we do not use any grammar in language modeling. The two syntactic constraints (i.e., a linkage is acyclic and planar) are considered in training data annotation, and are only captured implicitly in the resulting model. The promising empirical results that our dependence model achieved thus raise an interesting question: whether or not syntactic grammars capture exactly those term dependencies that we need for information retrieval? Our answer from the empirical results is probably no. We find in our experiments that many generated dependencies do not make sense from the syntactic point of view (e.g. the dependencies in Set B of Table 5), but the use of them reduces the entropy of the language model and results in improvements in information retrieval (e.g. #2, #31 and #35 queries in Table 5). In our experiments, among the 49 queries we used, the incorporation of term dependencies has a positive impact for 39 queries across all the six collections, and a negative impact for 8 queries. Some sample queries together with their dependencies detected are shown in Table 5.

Models	AvgP	% change over BM	% change over UG	# of dependencies
BM	18.62	--	--	--
UG	18.28	-1.83	--	7.2E5 unigram
CM	18.53	-0.48	+1.4	5.2E7
DM	19.64	+5.48*	+7.4*	2.5E7

Table 4. Comparison results on **ALL** collections. Results of the models CM and DM are obtained by re-ranking 1k-best lists generated using the BIR model. * indicates the difference is statistically significant according to t-test (*p*-value < 0.05).

6. Conclusion

We have presented a new dependence language modeling approach to information retrieval. In this approach, we introduce the linkage of a query as a hidden variable, which expresses the term dependencies within a sentence and forms an acyclic, planar, undirected graph. The approach then suggests generating a query from a document in two stages: first to generate the linkage, and then to generate each term in turn depending on other related terms according to the linkage. This is a general approach that covers several state-of-the-art language model approaches as special cases. We have also discussed how the proposed dependence model resolves the two problems of the classical dependence models: term dependency estimation and weight normalization. We demonstrated that our dependence model is applicable in the information retrieval system by (1) learning the linkage efficiently in an unsupervised manner; and (2) smoothing the model with different smoothing techniques. Our experiments on six standard TREC collections indicate the effectiveness of our dependence model: It outperforms substantially over both the classical probabilistic retrieval model and the state-of-the-art unigram and bigram language models.

7. Reference

[1] Buckley, D., Allan, J. and Salton, G. 1995. Automatic retrieval approaches using SMART: TREC-2. In: *Information Processing and Management*, 31, 315-326.

[2] Charniak, Eugene. 2001. Immediate-head parsing for language models. In: *ACL/EACL 2001*, pp.124-131.

[3] Chelba, Ciprian and Frederick Jelinek. 2000. Structured Language Modeling. In: *Computer Speech and Language*, Vol. 14, No. 4. pp 283-332.

#	Query	Set A	Set B	Impr. over UG
2	Status (of) nuclear proliferation treaties -- violations (and) monitoring.	(nuclear, proliferation) (proliferation, treaties) (status, monitoring)	(status, nuclear) (treaties, violation) (treaties, monitoring)	+19.96%
14	(What) (are) (the) different techniques used (to) create self-induced hypnosis?	(different, techniques) (self-induced, hypnosis) (techniques, used)	(used, create) (techniques, hypnosis)	-23.83%
31	(Should) (the) U.S. Government provide increased support (to) (the) National Endowment (for) (the) Arts?	(u.s., government) (government, provide) (provide, support) (national, endowment) (endow, art)	(provide, increased) (government, nation)	+28.80%
32	Reports (of) (and) evaluation (on) (the) near-death experience.	(reports, evaluation) (near-death , experience) (evaluation, experience)		0
34	(What) progress (has) (been) made (in) fuel cell technology?	(fuel, cell) (progress, made) (progress, technology)		+43.03%
35	(What) support (is) (there) (in) (the) U.S. (for) legalizing drugs?	(support, legalizing) (legalizing, drugs)	(u.s., drugs)	+26.68%
42	(How) (has) affirmative action affected (the) construction industry?	(affirmative, action) (action, affected) (construction, industry) (affected, industry)		+51.31%
46	(What) (is) (the) extent (of) U.S. arms exports?	(u.s., arms) (arms, exports) (extent, exports)		+33.72%

Table 5. Sample queries (where stop words are bracketed in **Query** column) together with their dependencies (links) and the impact on IR, where dependencies in Set A are syntactic meaningful in the sentence (i.e. query) and dependencies in Set B are not.

[4] Chelba, C, D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E. S. Ristad, R. Rosenfeld, A. Stolcke and D. Wu. 1997. Structure and performance of a dependency language model. In: *Processing of Eurospeech*, Vol. 5, pp 2775-2778.

[5] Collins, Michael John. 1996. A new statistical parser based on bigram lexical dependencies. In: *ACL 34*, pp. 184-191.

[6] Cooper. W. 1991. Some inconsistencies and misnomers in probabilistic information retrieval. In: *SIGIR 1991*, pp. 57-61.

[7] Croft, W. B. 1986. Boolean queries and term dependencies in probabilistic retrieval models. In: *JASIS*, 37(2): 71-77.

[8] Della Pietra, S., V. Della Pietra, J. Gillett, J. Lafferty, H. Printz and L. Ures. 1994. *Inference and estimation of a long-range trigram model.* Technical report CMU-CS- 94-188, Department of Computer Science, CMU.

[9] Fuhr, N. 1992. Probabilistic models in information retrieval. In: *The Computer Journal*, 35(3): 243-255.

[10] Harper, D. J. and C. J. van Rijsbergen. 1978. An evaluation of feedback in document retrieval using co-occurrence data. In: *Journal of Documentation*, 34: 189-216.

[11] Gao, Jianfeng, Jian-Yun Nie, Hongzhao He, Weijun Chen, and Ming Zhou. 2002. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In: *ACM SIGIR'02*, pp 183 – 190.

[12] Gao, Jianfeng and Hisami Suzuki. 2003. Unsupervised learning of dependency structure for language modeling. In: *ACL 2003*, pp. 521-528.

[13] Harman, D. K. 1995. Overview of the fourth Text REtrieval Conference (TREC-4). In: *TREC-4*, pp 1-24.

[14] Jelinek, Frederick. 1998. *Statistical methods for speech recognition.* The MIT Press, Cambridge, Massachusetts, London, England.

[15] Katz, S. M. 1987. Estimation of probabilities from sparse data for other language component of a speech recognizer. In: *IEEE transactions on Acoustics, Speech and Signal Processing*, 35(3): 400-401.

[16] Lewis, D. D. 1998. Naïve (Bayes) at forty: the independence assumption in information retrieval. In: *EMCL 1998*, pp. 4-15.

[17] Losee, R. M. 1994. Term dependence: truncating the Bahadur Lazarsfeld expansion. In: *Information Processing and Management*, 30(2): 293-303.

[18] Jones, K. S., S. Walker and S. Robertson. 1998. *A probabilistic model of information retrieval: development and status*. Technical Report TR-446, Cambridge University Computer Laboratory.

[19] Katz, S. M. 1987. Estimation of probabilities from sparse data for other language component of a speech recognizer. In: *IEEE transactions on Acoustics, Speech and Signal Processing*, 35(3): 400-401.

[20] Lafferty, J., Sleator, D. and Temperley, D. 1992. Grammatical trigrams: a probabilistic model of link grammar. In: *Proc. of the 1992 AAAI Fall Symposium on Probabilistic Approaches to Natural Language*.

[21] Lafferty, John and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In: *SIGIR'01*, pp. 111-119.

[22] Miller, D. H., Leek, T. and Schwartz, R. 1999. A hidden Markov model information retrieval system. In: *SIGIR'99*, pp. 214-221.

[23] Nallapati, R. and J. Allan. 2002. Capturing term dependencies using a language model based on sentence trees. In: *CIKM'02*, pp. 383-390.

[24] Ponte, J. and W. B. Croft (1998). A language modeling approach to information retrieval, In: *SIGIR'98*, pp. 275-281.

[25] Robertson, S. E. and S. Walker. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: *SIGIR 1994*, pp. 232-241.

[26] Robertson, S. E. and Walker, S. 2000. Microsoft Cambridge at TREC-9: Filtering track. In: *TREC-9*, pp. 361-368.

[27] Song, F. and Croft, B. 1999. A general language model for information retrieval. In: *CIKM'99*, pp. 316–321.

[28] Sparck Jones, K. 1998. What is the role of NLP in text retrieval? In: *Naturnal language information retrieval* (Ed. T. Strzalkowski), Dordrecht: Kluwer.

[29] Srikanth, M. And Srikanth, R. 2002. Biterm language models for document retrieval. In: *SIGIR 2002*, pp. 425-426.

[30] van Rijsbergen, C. J. 1977. A theoretical basis for the use of co-occurrence data in information retrieval. In: *Journal of Documentation*, 33(2): 106-119.

[31] Xu, J. and Croft, W. B. 2000. Improving effectiveness of information retrieval with local context analysis. In: *ACM Transactions on Information Systems*, 18(1): 79-112.

[32] Yuret, Deniz. 1998. *Discovery of linguistic relations using lexical attraction*. Ph.D. thesis, MIT, 1998.

[33] Zhai, Chengxiang, and John Lafferty. 2001. Two-stage language models for information retrieval. In: *SIGIR2002*, pp. 49-56.