Integrating Word Relationships into Language Models

Guihong Cao, Jian-Yun Nie, Jing Bai
Département d'Informatique et de Recherche Opérationnelle,
Université de Montréal
C.P. 6128, succursale Centre-ville
Montréal, Québec, H3C 3J7 Canada
{caogui, nie, baijing}@iro.umontreal.ca

ABSTRACT
In this paper, we propose a novel dependency language modeling approach for information retrieval. The approach extends the existing language modeling approach by relaxing the independence assumption. Our goal is to build a language model in which various word relationships can be integrated. In this work, we integrate two types of relationship extracted from WordNet and co-occurrence relationships respectively. The integrated model has been tested on several TREC collections. The results show that our model achieves substantial and significant improvements with respect to the models without these relationships. These results clearly show the benefit of integrating word relationships into language models for IR.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Retrieval Models – dependency language model, parameter setting, thesauri

General Terms
Algorithms, Theory, Experimentation, Performance

Keywords
Information Retrieval, Language Modeling, dependency, WordNet, Co-occurrence

1. INTRODUCTION
In recent years, language models for information retrieval (IR) have increased in popularity, due to their simplicity, clear probabilistic meaning, as well as efficiency and state-of-the-art performance [1, 6, 9, 12, 16]. The basic idea behind is to compute the conditional probability $P(Q|D)$, i.e., the probability of generating a query Q given the observation of a document D and the documents are ranked in descending order of this probability. A number of methods have been applied to compute this conditional probability. In most approaches, the computation is conceptually decomposed into two distinct steps: (1) Estimating the document model; (2) Computing the query likelihood using the estimated document model. When estimating the document model, the words in the document are assumed to be independent with respect to one another, leading to the so called "bag-of-word" model. However, from our own knowledge of natural language, we know that the assumption of term independence is a matter of mathematical convenience rather than a reality. For example, the words "computer" and "program" are not independent. A query requesting for "computer" might be well satisfied by a document about "program".

Some studies have been carried out to relax the independence assumption. This is generally done in two directions. The first one is data-driven, which tries to capture dependency among terms by statistical information derived from the corpus directly. For example, co-occurrences of terms may be used [1, 4, 5, 6, 10]. Term dependency can thus be integrated into language modeling. However, since the dependencies extracted from co-occurrences are blindly obtained from data, much noise can be introduced, which could undermine the retrieval effectiveness. Another direction is to exploit hand-crafted thesauri, such as WordNet [7, 8, 14]. WordNet has been used to recognize compound terms and dependencies among terms in these studies. The thesaurus is incorporated within classical information retrieval models, such as vector space model and probabilistic model [13]. To our knowledge, no one has yet tried to incorporate such a thesaurus within the language modeling framework.

In comparison with relationships extracted from corpora, manually built thesauri only contain manually validated relationships. They are thus less noisy (although ambiguous). In addition, many manually identified relationships can be hardly extracted automatically from corpora. Synonymy relationships are such example: it is difficult to automatically extract the relationship between "query" and "request", as a document would usually use only one term to designate the same object.

In this paper we propose a novel dependency model to incorporate both relationships of WordNet and co-occurrence within the language modeling framework for information retrieval. The possible advantage is twofold: On one hand, we can benefit from WordNet to cover related terms that cannot be identified automatically; on the other hand, we can rely on the manually recognized relationships that are supposed to be more precise, to complement the statistical relationships extracted from co-occurrences, while these latter insure generally a broad coverage of the possible relationships.

One of the difficulties for using WordNet in language modeling is that relations between terms in WorldNet are binary, i.e., one term is linked or not to another term. No weight is associated. When these relations are integrated into a language model, we will have to assign a probability to the link between two terms. A technique

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

relying on term co-occurrences will be used for this. Another problem concerns the combination of different types of relationships in a language model. We will deal with this problem through language model smoothing.

A series of experiments on standard TREC collections have been conducted to evaluate this method and the experimental results show that our approach is promising: by integrating each type of word relationship, we observe consistent improvements in retrieval effectiveness. This shows that manually built resources such as WordNet, as well as co-occurrence information, can be well incorporated into statistical language models to enhance IR.

The rest of the paper is organized as: Section 2 reviews previous work on relaxing the independence assumption and the utilization of WordNet in information retrieval. Section 3 presents our dependency language model to incorporate WordNet and co-occurrence relationships. Section 4 discusses the details for estimating model parameters. A serial of experiments on TREC collection are presented in Section 5, together with some further discussions. Section 6 summarizes the paper and suggests avenues for future work.

2. Previous Work

In the classical language modeling approaches [9, 12, 16] to IR, a multinomial model $P(w|d)$ over terms is estimated for each document d in the collection C to be indexed and searched. This model is used to assign likelihood to a user's query $q=q_1 q_2 \ldots q_n$. In most cases, each query term is assumed to be independent of the others, so that the query likelihood is estimated by $P(q \mid d) = \prod_{i=1}^{n} P(q_i \mid d)$. After the specification of a document prior $P(d)$, the posteriori probability of a document is given by

$$P(d \mid q) \propto P(q \mid d) P(d) \qquad (1)$$

The above probability is used to rank the documents.

As in speech recognition, a language model for information retrieval must be smoothed to adjust zero probability and small probabilities. Several smoothing strategies are discussed in [17]. "One of the main effects of smoothing is its robust estimation of common, content-free words that are typically treated as 'stop-words' in many information retrieval systems" [6]. However the classical language model approach for IR does not address the problem of dependence between words.

The term "dependence" may mean two different things: dependence between words within a query or within a document; dependence between query words and document words. Under the first meaning, one may try to recognize the relationships between words in a sentence (either in a document or in a query). In so doing, a sentence is no longer a bag of words. Rather, some dependence will be recognized between words. The approach proposed in [4] aims to recognize this type of dependence. Then a query is understood as a set of words, together with some links among them. These links are used as additional criteria to be verified by the documents to be retrieved.

Under the second meaning, dependence means any relationship that can be exploited during query evaluation, such as synonymy, in order to indirectly match a document with a query. For example, for a certain period of time, the document containing the word "Clinton" may well answer the query containing the term "president". The relationship between "Clinton" and "president" in this example is covered by the second meaning of dependence.

Both types of dependence are important for IR. In this study, we will concentrate on the second type.

To incorporate term relationships into the document language model, Berger and Lafferty [1] propose a translation model $t(q_i|w)$ for mapping a document term w to a query term q_i. In fact, the translation probability $t(q_i|w)$ describes the degree of link between the query term q_i and the document word w. With the translation model, the document-to-query model becomes

$$P(q \mid d) = \prod_{i=1}^{n} \sum_{w} t(q_i \mid w) P(w \mid d) \qquad (2)$$

Even though their model is more general than other language models, it is difficult to determine the translation probability $t(q_i|w)$ in practice. To solve this problem, Berger and Lafferty generate an artificial collection of "synthetic" data for training by assuming that a sentence is parallel to the paragraph that contains the sentence. This is indeed a variant use of co-occurrence information, although it is formulated in a different, statistical machine translation setting. Then the synthetic data have the same limitations as co-occurrence information, i.e. only some of the interesting relationships can be extracted (provided that the terms co-occur often enough), and the extracted relationships contain much noise.

Lafferty and Zhai [6] address this problem differently. They develop a more general model, Markov chain word translation model. It uses a random walk to derive the translation probability $t(q_i|w)$ from a set of documents in the collection. However, this probability is still estimated from term distribution or co-occurrences, without considering other term relationships explicitly. Jin and Hauptman [5] also propose a different method. They consider a document title as a possible query, and assume that the document is relevant to its title. Then they have a set of document-query pairs to train the translation model between document words and "query" terms

In all of above models, since $t(q_i|w)$ is trained from the document collection, it can only describe the link between terms in the document collection. Several problems arise. The first is that some desired relationships may not be extracted such as true synonymy relationships. The second problem is that virtually, any pair of terms that co-occur within the same document (or paragraph) could be considered to be related. As a consequence, the gain from relaxed independence assumption may not outweigh the loss due to the noise introduced.

The second family of approaches exploits term links stored in a hand-crafted thesaurus, such as WordNet. Voorhees [16] first exploits WordNet for query expansion. However, her experiments did not show any gain in retrieval effectiveness when queries are expanded by related terms. In the same vein, Liu et al. [9] use WordNet to disambiguate word senses of query terms and to expand queries. In their work, whenever the sense of the query term is determined, its synonyms, hyponyms, words from its definition and its compound words are considered for possible additions to the query.

Instead of using WordNet alone, Mandala et al. [8] use both WordNet and automatically constructed thesauri to expand queries. They build two thesauri from the corpus, a co-occurrence-based thesaurus and a predicate-argument-based

thesaurus [8], and assign a weight to each associated term pair in the thesauri to represent the degree of association. Since a relation between two terms in the WordNet has no weight, they assign it the average of the weights in co-occurrence-based thesaurus and predicate-argument-based thesaurus. They incorporate the three types of relationships within vector space model. Their experiments show that it is useful to combine WordNet with automatically construct thesauri for query expansion, and this results in improvements in retrieval effectiveness. Intuitively, manually and automatically establish relationships are complementary: the first ones are more precise but they have a limited coverage; the second ones have wider coverage but they contain much noise. By combining them in an appropriate way, we can benefit from the advantages of both. Our approach follows the same direction: we try to use both WordNet and relationships extracted from co-occurrences. However, an important difference is that we do not use ad hoc parameters to combine both types of relationships as Mandala et al. Instead, we will use a language modeling setting to combine them in a principled manner.

For a different problem – PP-attachment, [15] uses random walk models that also combine corpus statistics with other types of relationship such as synonymy relationships in WorldNet. To this respect, our approach follows the same direction.

3. A Dependency Model to Combine WordNet and Co-occurrence

The model proposed by Berger and Lafferty [1] provides a good general framework. In this paper, we will use a different formulation, which allows us to integrate different types of word relationships.

Given a query q and a document d, the query can be related directly, or they can be related indirectly through some word relationships. An example of the first case is that the document and the query contain the same words. In the second case, a document can contain a different word, but synonymous or related to the one in the query. In this case, the query can still be satisfied by the document. In order to take both cases into our modeling, we assume that there are two sources to generate a term from a document: one from a dependency model and another from a non-dependency model (which will be a unigram model in our case). Therefore, the likelihood of the query given a document can be expressed as follows:

$$\begin{aligned} P(q \mid d) &= \prod_{i=1}^{n} P(q_i \mid d) \\ &= \prod_{i=1}^{n} [P(q_i, \theta_D \mid d) + P(q_i, \theta_{\bar{D}} \mid d)] \\ &= \prod_{i=1}^{n} [P(q_i \mid d, \theta_D) P(\theta_D \mid d) + P(q_i \mid d, \theta_{\bar{D}}) P(\theta_{\bar{D}} \mid d)] \end{aligned} \quad (3)$$

where θ_D is the parameter of dependency model and $\theta_{\bar{D}}$ is the parameter of non-dependency model. $P(\theta_D \mid d)$ and $P(\theta_{\bar{D}} \mid d)$ are the probability to choose dependency model and non-dependency model respectively. As the non-dependency model tries to capture the direct generation of the query by the document (without considering any word relationships), we can model it by unigram document model, i.e.

$$P(q_i \mid d, \theta_{\bar{D}}) P(\theta_{\bar{D}} \mid d) = P_U(q_i \mid d) P(U \mid d)$$

where $P_U(q_i \mid d)$ is the probability of unigram model.

For the dependency model, we imagine a Markov process to generate a query term. First, we select a term in the document randomly. Second, a query term is generated based on the observed term. Here, term dependency enters into play. If the selected term is "computer" at first step, it is more likely to generate "cpu" than "water" in the second step. Therefore we have:

$$P(q_i \mid d, \theta_D) = \sum_{w \in d} P(q_i \mid w) P(w \mid d, \theta_D) \quad (4)$$

This formulation is equivalent to that of the translation model of Berger and Lafferty [1]. As for the translation model, we also have the problem of estimating the dependency between two terms, i.e. $P(q_i|w)$. Instead of considering only co-occurrence information as in the previous studies, we take a different approach here. We assume that some word relationships have been manually identified and stored in a linguistic resource (e.g. WordNet), and some other relationships have to be found automatically according to co-occurrences. Therefore, we have at least two different resources of word relationships. A word can be linked to another word through one of them. The global relationship between them can be made by combining both resources together. This combination can be achieved by a linear interpolation smoothing. Thus:

$$P(q_i \mid w) = \lambda P(q_i \mid L, w) + (1 - \lambda) P(q_i \mid \bar{L}, w) \quad (5)$$

where $P(q_i \mid L, w)$ is the conditional probability of q_i given w according to WordNet, which is called Link Model; $P(q_i \mid \bar{L}, w)$ is the probability that the link between q_i and w is achieved by other means (in our case, co-occurrences); λ is the interpolation factor, which can be viewed as mixture weight if Equation 5 is considered as a two-component mixture model. In our study, we only consider co-occurrence information beside WordNet. So $P(q_i \mid \bar{L}, w)$ is just the co-occurrence model. The estimations of all these models will be explained later.

For the simplicity of expression, we denote probability of link model as $P_L(q_i \mid w)$, i.e. $P_L(q_i \mid w) = P(q_i \mid L, w)$, and the co-occurrence model as $P_{CO}(q_i \mid w) = P(q_i \mid \bar{L}, w)$ hereafter. Substitute Equations 4 and 5 into 3, we obtain Equation 6.

$$\begin{aligned} P(q \mid d) &= \prod_{i=1}^{n} [P(q_i \mid d, \theta_D) P(\theta_D \mid d) + P_U(q_i \mid d) P(U \mid d)] \\ &= \prod_{i=1}^{n} [(\sum_{w \in d} P(q_i \mid w) P(w \mid d, \theta_D)) P(\theta_D \mid d) \\ &\qquad + P_U(q_i \mid d) P(U \mid d)] \\ &= \prod_{i=1}^{n} [\lambda P(\theta_D \mid d) \sum_{w \in d} P_L(q_i \mid w) P(w \mid d, \theta_D) \\ &\qquad + (1 - \lambda) P(\theta_D \mid d) \sum_{w \in d} P_{CO}(q_i \mid w) P(w \mid d, \theta_D) \\ &\qquad + P_U(q_i \mid d) P(U \mid d)] \end{aligned} \quad (6)$$

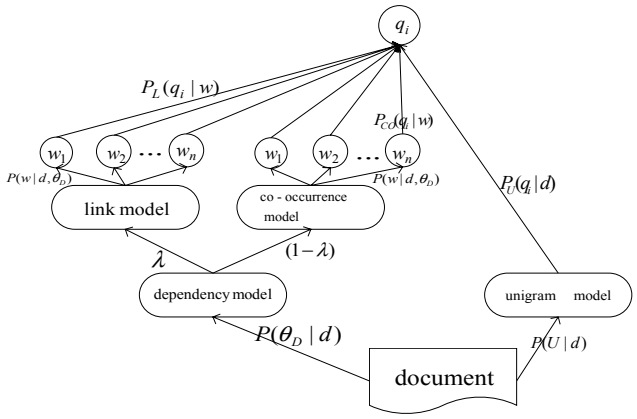

Figure 1: Bayesian Network for Generating a Query Term

This equation may seem complicated, but it incorporates a very intuitive idea: the relationship between a document word and a query word can be realized in several ways – direct connection when they are identical, indirect connection either through WordNet or through co-occurrences. Figure 1 gives a Bayesian network illustration of Equation 6.

The idea can become more obvious if we make some simplification in the formula. Let us define:

$$P_L(q_i \mid d) = \sum_{w \in d} P_L(q_i \mid w) P(w \mid d, \theta_D) \tag{7}$$

and

$$P_{CO}(q_i \mid d) = \sum_{w \in d} P_{CO}(q_i \mid w) P(w \mid d, \theta_D) \tag{8}$$

Equation 7 and 8 describe the probability of q_i in d from the link model and co-occurrence model respectively. Then Equation 6 can be put into the following simpler form:

$$\begin{aligned} P(q \mid d) = \prod_{i=1}^{n} [& \lambda P(\theta_D \mid d) P_L(q_i \mid d) \\ & + (1-\lambda) P(\theta_D \mid d) P_{CO}(q_i \mid d) \\ & + P_U(q_i \mid d) P(U \mid d)] \end{aligned} \tag{9}$$

Equation 9 clearly shows that we have indeed a three-component mixture model consisting of link model, co-occurrence model as well as unigram model. For each component, it has a mixture weight. Let $\lambda_L, \lambda_{co}, \lambda_U$ denote the respect weights of link model, co-occurrence model and unigram model, then Equation 9 can be rewritten as:

$$\begin{aligned} P(q \mid d) = \prod_{i=1}^{n} [& \lambda_L P_L(q_i \mid d) + \\ & \lambda_{CO} P_{CO}(q_i \mid d) + \lambda_U P_U(q_i \mid d)] \end{aligned} \tag{10}$$

where $\lambda_L = \lambda P(\theta_D \mid d)$, $\lambda_{CO} = (1-\lambda) P(\theta_D \mid d)$ and $\lambda_U = P(U \mid d)$. The above equation defines the general principle of our approach, which places the approach of Mandala et al. into a language modeling framework.

In the above formulation, we consider only one type of relationship in WordNet. Indeed, several types of relationship are stored in WordNet, for example, synonymy relation, hypernymy relation, and so on. Different types of relation should not play the same role. It is more reasonable to separate the link model into several sub-models, each corresponding to a specific type of relation. For information retrieval, the most important terms are nouns, so we concentrate on three relations related to nouns: synonym, hypernym and hyponym. Let $P_{SYN}(q_i \mid d)$, $P_{HYPE}(q_i \mid d)$ and $P_{HYPO}(q_i \mid d)$ denote the synonym model, hypernym model and hyponym model respectively. Then equation 10 can be extended to:

$$\begin{aligned} P(q \mid d) = \prod_{i=1}^{n} [& \lambda_1 P_{SYN}(q_i \mid d) + \lambda_2 P_{HYPE}(q_i \mid d) + \\ & \lambda_3 P_{HYPO}(q_i \mid d) + \lambda_4 P_{CO}(q_i \mid d) + \lambda_5 P_U(q_i \mid d)] \end{aligned} \tag{11}$$

where λ_i $(i = 1,...,5)$ are the mixture weights of the five models. In our discussion, we will refer to the dependency model with non-separated link model (Eq. 10) as ***NSLM*** and the one with separated link model (Eq. 11) as ***SLM*** hereafter. Now the remaining problem is to estimate the parameters in the models, such as the conditional probabilities, the weights of various models etc. We will discuss these problems in the next section.

4. Parameter estimation

In NSLM, 7 terms have to be estimated: $P_U(q_i|d)$, $P(w|d,\theta)$, $P_L(q_i|w)$, $P_{co}(q_i|w)$, and the three mixture weights. In SLM, $P_L(q_i|w)$ is split into three sub-elements, so is the associated mixture weight. So the number amounts to 11. In the following, we only describe the estimation of the parameters in NSLM. Those in SLM can be estimated in a similar way.

4.1 Estimating conditional probabilities

The unigram model $P_U(w_i|d)$ can be estimated using any existing method. In our case, we use the ***MLE*** estimation, smoothed by interpolated absolute discount [17], that is:

$$P_{abs}(w_i \mid d) = \frac{\max(c(w_i;d) - \delta, 0)}{|d|} + \frac{\delta \mid d \mid_u}{|d|} P_{MLE}(w_i \mid C) \tag{12}$$

where δ is the discount factor, $|d|$ is the length of the document, $|d|_u$ is the count of unique term in the document, and $P_{MLE}(w_i|C)$ is the maximum likelihood probability of the word in the collection C. This smoothing method is chosen among a set of other smoothing methods (such as Jelinek-Mercer smoothing and Dirichlet smoothing [17]) because we found that this smoothing showed most stable performance in our experiments.

For $P(w \mid d, \theta_D)$ - the probability of w in document d according to dependency model(s), it can be approximated by the maximum likelihood probability $P_{MLE}(w|d)$. This approximation is motivated by the fact that the word w is primarily generated from d in a way quite independent from the model θ_D.

The key problem now is the estimation of $P_L(w_i|w)$ – the probability of link between two words according to WordNet. Notice that WordNet does not provide any weight to relations. A naïve approximation would be to assign the relationship a binary weight (or possibly with a normalization). However, this could not reflect correctly the strength of the connection between the words. Instead, we will rely on the text collection to determine the probability by counting the co-occurrences of these words in the collection. In addition, we impose a condition on the co-occurrences: the words should co-occur within a window W of certain size. This approach uses a similar idea to that of Mandala et al [8].

As many pairs of words in the vocabulary have no link in WordNet, $P_L(w_i|w)$ can not be calculated by the relative frequency of co-occurrences alone. Smoothing has to be used. We tried four smoothing methods, Jelinek-Mercer, Dirichlet, Absolute Discount and Kneser-Ney as well as two smoothing strategies, backoff and interpolation [2]. It turns out that interpolated Absolute discount and Kneser-Ney have the best performance, which is consistent with Chen and Goodman's conclusion [2].

Equation 13 defines our estimation of $P_L(w_i|w)$ by interpolated Absolute discount:

$$P_L(w_i \mid w) = \frac{\max(c(w_i, w \mid W, L) - \delta, 0)}{\sum_{w_j} c(w_j, w \mid W, L)} + \frac{c(*, w \mid W, L)\delta}{\sum_{w_j} c(w_j, w \mid W, L)} P_{add-one}(w_i \mid W, L)$$
$$P_{add-one}(w_i \mid W, L) = \frac{\sum_{j=1}^{|v|} c(w_i, w_j \mid W, L) + 1}{\sum_{i=1}^{|v|}\sum_{j=1}^{|v|}(c(w_i, w_j \mid W, L) + 1)} \quad (13)$$

where w_i and w are assumed to have a relationship in WordNet, $C(w_i, w|W, L)$ is the count of co-occurrences of w_i with w within the predefined window W, and $C(*, w|W, L)$ is the number of unique terms which have a relationship with w_i in WordNet and co-occur with it in W.

Notice that the above estimation is similar to a biterm language model [14], in which word co-occurrences are considered without word order. The difference is that we only consider the pairs of words connected in WordNet.

The estimation of the components of the co-occurrence model $P_{co}(w_i|d)$ is similar to those of the link model $P_L(w_i|d)$ except that when counting the co-occurrence frequency, the requirement of having a link in WordNet is removed. It can be calculated by Equation 14, also smoothed by interpolated Absolute discount.

$$P_{CO}(w_i \mid w) = \frac{\max(c(w_i, w \mid W) - \delta, 0)}{\sum_{w_j} c(w_j, w \mid W)} + \frac{c(*, w \mid W)\delta}{\sum_{w_j} c(w_j, w \mid W)} P_{add-one}(w_i \mid W)$$
$$P_{add-one}(w_i \mid W) = \frac{\sum_{j=1}^{|v|} c(w_i, w_j \mid W) + 1}{\sum_{i=1}^{|v|}\sum_{j=1}^{|v|}(c(w_i, w_j \mid W) + 1)} \quad (14)$$

The estimation of synonym model, hypernym model and hyponym model in SLM follows the same way, except that each type of relation is considered separately in a sub-model.

4.2 Estimating mixture weights

In this section we introduce an EM algorithm to estimate the mixture weights in NSLM. Because NSLM is a three-component mixture model, the optimal weights should maximize the likelihood of the queries [18]. For each query q in the dataset (in our case, we use TREC topics 51-100), let $\theta_q = [\lambda_L, \lambda_{CO}, \lambda_U]$ be the mixture weights, we then have:

$$\theta_q^* = \arg\max_{\theta_q} \log \sum_{i=1}^{N} \pi_i \prod_{j=1}^{m} [\lambda_U P_U(q_j \mid d_i) + \lambda_L P_L(q_j \mid d_i) + \lambda_{CO} P_{CO}(q_j \mid d_i)] \quad (15)$$

where N is the number of documents in the dataset, and m is the length of query q. $\{\pi_i\}_{i=1}^N$ acts as the prior probability with which to choose the document to generate the query. Thus the query is generated from a mixture of N document models with unknown mixing weight $\{\pi_i\}_{i=1}^N$. Note that leaving $\{\pi_i\}_{i=1}^N$ unfixed is important, because what we really want is not to maximize the likelihood of generating the query from every document in the collection. Instead this maximization is modulated by $\{\pi_i\}_{i=1}^N$ which assign some weight to different documents according to their relatedness to the query: the more a document model can generate the query, the more we want to maximize it. With $\{\pi_i\}_{i=1}^N$ as free parameters to be estimated, we would indeed allocate higher weight to documents that generate the query well; presumably, these documents are also more likely to be relevant.

The method is similar in principle to pseudo-relevance feedback, which assumes the top n documents to be relevant to the query. Ranking at top level is equivalent to having a high weight in our case. [18] employs the same method to learn mixture weights.

However, there arises another problem. Some documents having high weights are not truly relevant to the query. They contain noise. To account for the noise, we further assume that there are two distinctive sources to generate the query, one is the relevant documents, another is a noisy source, which is approximated by the collection C. Then Equation 15 is rewritten as:

$$\theta_q^* = \arg\max_{\theta_q} \log \left\{ \begin{array}{l} (1-\alpha)\sum_{i=1}^{N} \pi_i \prod_{j=1}^{m} [\lambda_U P_U(q_j \mid d_i) \\ \quad + \lambda_L P_L(q_j \mid d_i) + \lambda_{CO} P_{CO}(q_j \mid d_i)] \\ + \alpha \prod_{j=1}^{m} [\lambda_U P_U(q_j \mid C) \\ \quad + \lambda_L P_L(q_j \mid C) + \lambda_{CO} P_{CO}(q_j \mid C)] \end{array} \right\} \quad (16)$$

where α is the weight of the noise, $P_U(q_j|C)$, $P_L(q_j|C)$ and $P_{CO}(q_j|C)$ are respectively unigram model, link model and co-occurrence model built from the collection. Here we fix α at a non-zero value, otherwise it would become close to zero because in that way, the documents would have higher likelihood and Equation 16 would reduce to Equation 15. In fact, the role of α is to add some robustness facing to the noise of the training data. In our experiments, α is set to 0.3. With this setting, the hidden $\{\pi_i\}_{i=1}^N$ and θ_q can be estimated using the EM algorithm [3]. The update formulas are as follows (we do not give their derivation here due to space limit):

$$\pi_i^{(r+1)} = \frac{\pi_i^{(r)} \prod_{j=1}^{m} [\lambda_U^{(r)} P_U(q_j \mid d_i) + \lambda_L^{(r)} P_L(q_j \mid d_i) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid d_i)]}{\sum_{i=1}^{N} \pi_i^{(r)} \prod_{j=1}^{m} [\lambda_U^{(r)} P_U(q_j \mid d_i) + \lambda_L^{(r)} P_L(q_j \mid d_i) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid d_i)]} \quad (17)$$

and

$$\lambda_U^{(r+1)} = \frac{1}{m} \frac{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)} \lambda_U^{(r)} P_U(q_j \mid d_i) + \alpha \lambda_U^{(r)} P_U(q_j \mid C)}{\{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)} [\lambda_U^{(r)} P_U(q_j \mid d_i) + \lambda_L^{(r)} P_L(q_j \mid d_i) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid d_i)] + \alpha [\lambda_U^{(r)} P_U(q_j \mid C) + \lambda_L^{(r)} P_L(q_j \mid C) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid C)]\}}$$

$$\lambda_L^{(r+1)} = \frac{1}{m} \frac{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)} \lambda_L^{(r)} P_L(q_j \mid d_i) + \alpha \lambda_L^{(r)} P_L(q_j \mid C)}{\{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)} [\lambda_U^{(r)} P_U(q_j \mid d_i) + \lambda_L^{(r)} P_L(q_j \mid d_i) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid d_i)] + \alpha [\lambda_U^{(r)} P_U(q_j \mid C) + \lambda_L^{(r)} P_L(q_j \mid C) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid C)]\}} \quad (18)$$

$$\lambda_{CO}^{(r+1)} = \frac{1}{m} \frac{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)} \lambda_{CO}^{(r)} P_{CO}(q_j \mid d_i) + \alpha \lambda_{CO}^{(r)} P_{CO}(q_j \mid C)}{\{(1-\alpha)\sum_{i=1}^{N} \pi_i^{(r)} [\lambda_U^{(r)} P_U(q_j \mid d_i) + \lambda_L^{(r)} P_L(q_j \mid d_i) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid d_i)] + \alpha [\lambda_U^{(r)} P_U(q_j \mid C) + \lambda_L^{(r)} P_L(q_j \mid C) + \lambda_{CO}^{(r)} P_{CO}(q_j \mid C)]\}}$$

The five mixture weights in SLM can also be estimated by EM algorithm in a similar way. We do not list the formulas here.

To terminate the EM iteration, we set a threshold on the change of the log-likelihood of the query: If the change is less than the threshold, EM algorithm stops. In our experiments, we find that EM for NSLM converges very quickly: It usually converges after about 5 iterations. For SLM, it converges after 10 iterations.

The above algorithm is very similar to the one proposed by Zhai and Lafferty [18] except that we introduce the noisy source into our model. In our experiments, it turns out that setting α to a non-zero value is slightly better than setting it to zero, which shows that it is beneficial to take into account the noise source in the model in an appropriate way.

5. Experiments

5.1 Experimental setting

Table 1. Statistics of Data Set

Coll.	Description	Size (MB)	# Doc.	Vocab. Size
WSJ	*Wall Street Journal* (1990-92), Disk 2	242	74,520	121,944
AP	*Associate Press* (1988-90), Disks 2&3	729	242,918	245,748
SJM	*San Jose Mercury News* (1991), Disk 3	287	90,257	146,512
Total		1,258	407,695	514,204

We evaluated our model described in the previous sections using three different TREC collections – WSJ, AP and SJM. Some statistics are shown in Table 1. All documents have been processed in a standard manner: terms were stemmed using the Porter stemmer and stopwords were removed. The queries are TREC 51-100. We used the title field and description field of the topics. These queries contain about 15-18 words. The document set comes from the TREC disks 2 and 3.

The version of WordNet we use 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 link model, the pool is used to determine whether the terms have a link. As we do not consider explicitly compound terms, all the compound terms in WordNet are decomposed into their component words.

The baseline of our experiment is the unigram model smoothed by interpolated Absolute discount. In the statistical language modeling approach for IR, there are some free parameters be estimated, for instance, the discount δ. In our experiments, we empirically set the parameters for unigram model by trial and error, and the parameters of the dependency model are blindly set at the same values as in the unigram model. So our dependency model is not tuned to its best. Even though, our dependency model outperforms the baseline substantially.

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.

5.2 Experimental Results

We used Lemur3.0 [11] to carry out experiments. For our purpose, Lemur has been extended to support our dependency language model. The baseline results are obtained directly by using Lemur. Table 2 shows the results of the first group of experiments, in which we compare unigram model with two kinds of dependency models, NSLM and SLM.

We see that dependency model (both NSLM and SLM) outperforms the unigram model over the three datasets. Specifically, the improvement on AP is greater than 10% and the other two datasets are above 5%. The improvement of WSJ and AP are statistically significant (at the level of *p-value* less than 0.05). The dependency model also performs well in recall. For each dataset, it retrieves more relevant documents than the unigram model. This is because unigram model only uses direct matching between document and query while our model has the capability to expand the document so as to match different query words. The increase in recall confirms this expansion effect.

We can also observe the difference between NSLM and SLM. It can be seen that differentiating the relations in WordNet (SLM) is better than mixing them (NSLM). We will further discuss this in section 5.4.

5.3 The role of link model

Compared with previous work on dependency language model, the difference of our work is the introduction of link model based on WordNet. So we conducted experiments to investigate the role of the latter. Table 3 shows the results. Here UM, LM and CM denote unigram model, link model and co-occurrence model respectively. From the table we can see that even though we cannot obtain good results using LM alone (which is expectable),

Table 2: Comparison between Unigram Model and Dependency Model.

Coll.	Unigram Model		Dependency Model					
			NSLM			SLM		
	AvgP	Recall	AvgP	% change	Recall	AvgP	% change	Recall
WSJ	0.2466	1659/2172	0.2597	+5.31*	1704/2172	***0.2623***	+6.37*	1719/2172
AP	0.1925	3289/6101	0.2128	+10.54**	3523/6101	***0.2141***	+11.22**	3530/6101
SJM	0.2045	1417/2322	0.2142	+4.74	1572/2322	***0.2155***	+5.38	1558/2322

AvgP is the non-interpolated average precision. * and ** indicate that the difference is statistically significant according to t-test at the level of *p-value* < *0.05* and *p-value* < *0.01*.

Table 3: Different combinations of unigram model, link model and co-occurrence model

Model	WSJ		AP		SJM	
	AvgP	**Recall**	**AvgP**	**Recall**	**AvgP**	**Recall**
UM	0.2466	1659/2172	0.1925	3289/6101	0.2045	1417/2322
CM	0.2205	1700/2172	0.2033	3530/6101	0.1863	1515/2322
LM	*0.2202*	*1502/2172*	*0.1795*	*3275/6101*	*0.1661*	*1309/2322*
UM+CM	0.2527	1700/2172	0.2085	3533/6101	0.2111	1521/2322
UM+LM	0.2542	1690/2172	0.1939	3342/6101	0.2103	1558/2332
UM+CM+LM	***0.2597***	***1704/2172***	***0.2128***	***3523/6101***	***0.2142***	***1572/2322***

it is always helpful to incorporate it in the model: whenever LM is incorporated, we observe some improvements. The combination of all the three models (UM+CM+LM) always outperforms significantly other partial combinations. The results confirm our hypothesis that the relations contained in WordNet (link model) can well complement the statistical relationships extracted from co-occurrences and enhance the retrieval performance. The poor performance obtained when using LM alone may be explained by the fact that LM is too small to include enough information. In fact, in our experiments, LM is usually less than 10 MB, while CM is usually 40 times larger than it.

5.4 The role of different relations in the WordNet

In section 5.2, we draw the conclusion that separating the relations in WordNet and treating them differently results in better effectiveness than treating them without any differentiation. In this section, we investigate the impact of different relations on retrieval effectiveness. Table 4 shows the average weights of different components of SLM over all queries. Here SM, HEM and HOM denote the synonym, hypernym and hyponym models respectively. These weights indicate, to some degree, the contribution of each component to the global performance of the model.

We can see that the relations of WordNet have different contributions in various collections. This may indicate that these relations may be useful for IR at different degrees in different areas.

Table 4: Average weight for different relations over all queries

Model	WSJ	AP	SJM
UM	0.3564	0.3006	0.4858
CM	0.1480	0.5282	0.1588
SM	0.1657	0.0883	0.1392
HEM	0.1745	0.0491	0.0963
HOM	0.1649	0.0338	0.11968
Total	1.0	1.0	1.0

It is also interesting to observe the correlation between the weights assigned to WordNet relations and the increases that we can obtain when these relations are incorporated (Table 3). For WSJ, we observe quite strong weights for WordNet relations, and we also observe a quite large improvement of UM_LM over UM in Table 3. On the other hand, on AP, the weights assigned to WordNet relations are very weak. We also observe only a marginal of performance change from UM to UM_LM in table 3 on this collection. This correlation tends to show that the suitability of WordNet to a particular document collection can be automatically determined by the parameter tuning process. In other words, the tuning process is able to determine the appropriate weights for WordNet relations according to their suitability to the area of the documents. Pushing our observation a step further: with an appropriate tuning process, the incorporation of WordNet in our model could not harm retrieval effectiveness. This observation also applies to other resources such as co-occurrence information. Thus, it could be helpful to incorporate in a retrieval model as many resources of different kinds as possible.

6. Conclusion and future work

In this paper, we propose a novel dependency language modeling approach for information retrieval. In this approach we integrate word relationships into the language modeling framework. Relationships come from two sources: one is from co-occurrences of terms in the dataset and the other is from WordNet.

The advantage of incorporating co-occurrence information in language modeling has been confirmed by several previous studies [4, 5, 6]. However, no previous study has investigated a different type of manually defined relationship in language modeling. Our study is motivated by the intuition that the addition of a manual resource can have two advantages: On one hand, we can benefit from such a resource to cover related terms that cannot be discovered automatically; on the other hand, we can rely on the manually recognized relationships that are supposed to be more precise to complement the statistical relationships extracted from co-occurrences. Our experiments confirm this intuition: whenever WordNet is incorporated, we observe some consistent (although variable) increase in retrieval effectiveness. The same observation is also true for the incorporation of co-occurrence information. Then our global conclusion of this study is that it is always better to incorporate more resources of different kinds into a language model for IR, provided that there is an appropriate training process to determine the parameters of the model correctly.

In this paper, we used EM algorithm to train the parameters. This method worked well for our experiments.

The co-occurrence model used in this study is not sophisticated. It is derived by observing term co-occurrences within texts, without making any filtering of noise. It would be interesting to integrate other more sophisticated methods such as those proposed in [1], [5] and [6] in our link model.

In this paper, we only studied the relationships between query words and document words. One interesting extension is to also consider the dependencies between query words or between document words [4]. This can help solve the problem of ambiguity. A related area is to consider not only single words, but also compound terms in language modeling. This can also create a more precise representation of document contents.

In our work, we assumed that word dependencies are independent of document. This is a simplification assumption. In reality, there is some dependence. So another interesting research direction is to make the dependencies between words dependent on specific document. However, a serious problem concerns the large number of parameters to estimate. We will investigate this issue in the future.

7. REFERENCES

[1] Berger, A. and Lafferty, J. (1999). Information retrieval as statistical translation. *In Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222-229.

[2] Chen, S.F. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. *Tech. Rep. TR-10-98, Harvard University.*

[3] Dempster, A.P, Laird, N. M., and Rubin, D. B. (1997). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*,39:1-38.

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

[5] Jin, R., Hauptmann, A.G., and Zhai, CX. (2002). Title Language Model for Information Retrieval. *In Proceedings of the 2002 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42-48.

[6] Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. *In Proceedings of the 2001 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111-119.

[7] Liu, S., Liu, F., Yu, C., and Meng, W., (2004). An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. . *In Proceedings of the 2004 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 266-272.

[8] Mandala, R., Tokunaga, T., and Tanaka, H. (1998). Ad Hoc Retrieval Experiments Using WordNet and Automatically Constructed Theasuri. *In Proceedings of the seventh Text REtreival Conference*, pages 475-481.

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

[10] Nallapati, R. and Allan, J. (2002). Capturing Term Dependencies using a Language Model based on Sentence Trees. *In Proceedings of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA, 2002*, pages 383-390.

[11] Ogilvie, P. and Callan, J. (2001). Experiments using the lemur toolkit. *In Proceedings of the Tenth Text Retrieval Conference (TREC-10)*, pages 103–108.

[12] Ponte, J. and Croft, W.B. (1998). A language modeling approach to information retrieval. *In Proceedings of the 1998 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275-281.

[13] Robertson, S.E., Van, Rijsbergen, C.J., and Poter, M.F., (1981). Probabilistic models of indexing and searching. *In Information Retrieval Research, R.N. Odd et al, Eds. Butterworths*, pages 35-56.

[14] Srikanth, M. and Srikanth, R. (2002). Biterm language models for document retrieval. *In Proceedings of the 2002 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 425-426.

[15] Toutanova, K., Manning, C.D. and Ng, A.Y. (2004). Learning random walk models for inducing word dependency distribution, *In Proc. of the 21st Int. Conf. on Machine Learning*, Banff.

[16] Voorhees, E. (1994). Query Expansion Using Lexical-Semantic Relations *In Proceedings of the 1994 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61-69.

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

[18] Zhai, C, and Lafferty, J. (2002). Two-stage language models for information retrieval. *In Proceedings of the 2002 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49-56.