Inferential Language Models for Information Retrieval

JIAN-YUN NIE, GUIHONG CAO, and JING BAI
University of Montreal

Language modeling (LM) has been widely used in IR in recent years. An important operation in LM is smoothing of the document language model. However, the current smoothing techniques merely redistribute a portion of term probability according to their frequency of occurrences only in the whole document collection. No relationships between terms are considered and no inference is involved. In this article, we propose several inferential language models capable of inference using term relationships. The inference operation is carried out through a semantic smoothing either on the document model or query model, resulting in document or query expansion. The proposed models implement some of the logical inference capabilities proposed in the previous studies on logical models, but with necessary simplifications in order to make them tractable. They are a good compromise between inference power and efficiency. The models have been tested on several TREC collections, both in English and Chinese. It is shown that the integration of term relationships into the language modeling framework can consistently improve the retrieval effectiveness compared with the traditional language models. This study shows that language modeling is a suitable framework to implement basic inference operations in IR effectively.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models, relevance feedback*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language models*

General Terms: Experimentation, Theory

Additional Key Words and Phrases: Query expansion, document expansion, inference, inferential model

1. INTRODUCTION

The goal of information retrieval (IR) is to find the documents relevant to a query. By relevant, we usually mean that the retrieved documents should be about the same (or at least similar) topic as the query. This does not mean that it is a necessary and sufficient condition that a relevant document contains all the keywords of the query. For example, it is possible that a document about algorithm complexity does not contain the keyword programming, but is

Authors' addresses: DIRO, University of Montreal, CP 6128, succursale Centreville, Montreal, Quebec H3C 3J7 Canada; email: {nie, caogui, baijing}@iro.umontreal.ca.

nevertheless relevant to a query on programming; a document containing the keywords house and construction in a debate of the House of Commons may not be relevant to the query on house construction. These problems are well documented and are also referred to as the synonymy and polysemy problems.

Despite the problems just mentioned, the current state-of-the-art in IR implements the ranking score as a matching function between the words (or terms) in the document and the query: the more a document shares the terms with the query, the higher this document will be ranked in the retrieval result. Although this term matching function can find some relevant documents because most relevant documents indeed share some of the terms of the query, it usually also retrieves much noise and misses many relevant documents.

To deal with these problems, various approaches have been proposed. For example, instead of using words, one can create a new representation space from them that may better correspond to semantic dimensions of documents and queries. Latent semantic indexing [Deerwester et al. 1990] is such an example. One can also incorporate semantic relations between terms stored in a thesaurus to take into account the synonyms (or related terms). Typically, this can be used in query expansion in order to add related terms into the original query so that it has better coverage of the relevant documents. The utilization of such a resource is well motivated, because this would correspond to a truly semantic IR that uses inference based on term relationships. In practice, however, it is difficult to use it in an appropriate way. For example, in the experiments of Voorhees [1994], it has been shown that using WordNet [Miller 1990] does not improve the retrieval effectiveness. Although this result has been improved since then (indeed, some recent studies [Cao et al. 2005 and Mandala et al. 1998] show that we can increase the retrieval effectiveness using WordNet), a general framework appropriate for its implementation is still lacking.

As Nie [2003] shows, query expansion is a particular case of inferential IR. Research Projects on general inferential IR started long ago (e.g., Croft et al. [1988]; van Rijsbergen [1986]). Many of these studies concentrate on the definition of new logical models capable of making inferences [Huibers et al. 1996; Nie 1990; Bruza and Huibers 1996]. The basic idea is to consider the retrieval process as an inference process, that is, to infer whether there is a relevance relationship between the document and the query. This idea has been formulated in different ways, for example, as a Bayesian network [Turtle and Croft 1990], or in logic terms [van Rijsbergen 1986]. In the latter case, many of the proposed approaches either cannot scale up or they can hardly be implemented efficiently.

In recent years, language modeling has been successfully applied to IR. This new family of approaches has attracted a great number of investigations because of their competitive experimental results as well as well-founded theoretical background. However, current LM approaches are unable to make logical inference. The central operation in them is model smoothing, which combines two (or more) term distributions, without using relationships between terms. Nevertheless, Berger and Lafferty [1999]; Cao et al. [2005]; Bai et al. [2005] have shown that one can integrate relationships between terms (term relationships) into LM. In this article, we argue and demonstrate that LM is an

appropriate framework to implement basic inference operations, and this results in an effective and efficient inferential model for IR.

In the rest the article, we will first review some of the previous approaches on inferential IR and LM. Then we will propose a new inferential IR model based on LM. Our implementation approach is still to use smoothing either on the document model or on the query model, but the smoothing process is done not only according to the collection model, but also according to term relationships. Several implementation details will then be described. The experiments of these models will be reported. These experiments are carried out on several collections (both in English and Chinese), showing that by integrating term relationships, we can obtain consistent and significant improvements in retrieval effectiveness.

2. LANGUAGE MODELING APPROACHES TO IR

Let us first review some typical approaches to IR based on LM.

The basic idea of LM in IR was proposed in Ponte and Croft [1998]. Given a document D, a statistical language model (a unigram model) is constructed for D, $P(.|D)$. For a query Q, this document is ranked according to the probability $P(Q|D)$—the probability of generating Q by the document language model. It is assumed that a query Q is composed of a sequence of words $q_1 q_2 \cdots q_n$,which are considered to be independent. So $P(Q|D)$ can be estimated as follows:

$$P(Q|D) = P(q_1 q_2 \cdots q_n|D) = \prod_{i=1}^{n} P(q_i|D). \tag{1}$$

Another formulation often used in IR is to rank a document according to the negative Kullback-Liebler divergence (KL-divergence):

$$\begin{aligned} -KL(Q||D) &= -\sum_{q_i \in Q} P(q_i|Q) \log \frac{P(q_i|Q)}{P(q_i|D)} \\ &= \sum_{q_i \in Q} P(q_i|Q) \log P(q_i|D) - \sum_{q_i \in Q} P(q_i|Q) \log P(q_i|Q). \end{aligned}$$

As the last element in this formula only depends on Q, it does not affect the ranking of documents. So

$$Score(D, Q) = \sum_{q_i \in Q} P(q_i|Q) \log P(q_i|D). \tag{2}$$

It is easy to see a close relationship between formulas (1) and (2). In fact, if we simply use an ML estimation for $P(q_i|Q)$ in formula (2), then:

$$\begin{aligned} Score(D, Q) &= \sum_{q_i \in Q} \frac{C(q_i; Q)}{|Q|} \log P(q_i|D) \propto \sum_{q_i \in Q} C(q_i; Q) \log P(q_i|D) \\ &= \sum_{q_i \in Q} \log P(q_i|D)^{C(q_i;Q)} = \log \prod_{i=1}^{n} P(q_i|D) \propto P(Q|D). \end{aligned}$$

Indeed, when KL-divergence is used in IR the query model is usually not smoothed, and we only use maximum likelihood (ML) estimation $P_{ML}(q_i|Q)$. This is due to the fact that if the query model is smoothed as a document

model, many terms will have nonzero probability, and, as a consequence, the query evaluation process becomes very expensive.

In the previous equations, $P(q_i|D)$ cannot be estimated by the following maximum likelihood estimation:

$$P_{ML}(q_i|D) = \frac{c(q_i; D)}{|D|},$$

where $c(q_i;D)$ is the occurrence count of q_i in D, and $|D|$ is the size of D. This is because some query terms may not appear in a document D, and this would lead $P(Q|D)$ to 0, a result that we do not desire in IR. In practice, even when a document does not contain some of the query words, it still can be relevant, and it can be retrieved. Therefore, a relaxation is necessary. Smoothing is used to relax the constraint that all the query words should appear in a document. Smoothing tries to attribute nonzero probabilities to terms that do not appear in a document. Another interpretation of smoothing is to consider a document as the result of a sampling process; some terms are selected, and some others are not. Among the missing terms, some should have been selected. So by smoothing, we try to recover those missing terms that should have been selected.

A common smoothing method is the following Jelinek-Mercer smoothing:

$$P_{JM}(q_i|D) = \lambda P_{ML}(q_i|D) + (1 - \lambda)P_{ML}(q_i|C).$$

This smoothing method makes a linear combination of two LMs, one created from the document and another from the whole document collection C. λ is a smoothing factor which can be tuned empirically or through an automatic process such as expectation maximization (EM) [Dempster et al. 1997].

Many other smoothing techniques have been developed in statistical LM [Chen and Goodman 1998], and several have been successfully used in IR. Zhai and Lafferty [2001a] made an empirical study of the effects of smoothing on IR effectiveness. It turns out that, by smoothing, one can naturally incorporate the IDF factor commonly used in IR.

However, the fact of attributing a nonzero probability to a missing term is by no means an inference. As we can see in the Jalinek-Mercer smoothing, this change is only due to a combination with another model built on the whole document collection. No term relationship is used. For example, given a document about natural language processing (NLP) and a collection containing documents of various topics, it is very possible that after smoothing, the term artificial intelligence (AI) would have similar probability to the terms tourism and shopping in that document: all the latter terms may be absent from the original document on natural language processing. Therefore, their probability may be a small value gained through probability redistribution during smoothing. If the whole collection is general (not specific to computer science), then it is possible that these terms have comparable probabilities after smoothing. Intuitively, an inference process would attribute a higher probability to AI than to tourism and shopping because of its strong relation to NLP. This example clearly shows that the previous smoothing process is not a logical inference.

Despite this, the smoothing methods proposed so far have the merit of being robust to noise which is inherent in an IR environment. By *noise* we mean

the fact that a document contains not only words strongly related to its topics, but also words unrelated to the topics. For example, a document talking about airline traffic in America can use words such as Japan, car, etc., which are not related to the topic. Therefore, an IR method cannot take for granted that if a term is present in a document, it describes a relevant piece of semantics of the document. Neither can one consider that a missing term means that the corresponding semantics is also absent from the document. Through the relaxation effect, the smoothing methods allow us to obtain a query evaluation process more resistant to noise.

In an attempt to integrate term relationships, Berger and Lafferty [1999] proposed a translation model for IR in which term relationships (formulated as translation relationships) are integrated into the evaluation function as follows:

$$P(q_i|D) = \sum_j t(q_i|w_j)P(w_j|D),$$

where $t(q_i|w_j)$ is the translation probability of w_j to q_i. In Berger and Lafferty [1999], this relationship is estimated for pairs of words in the same language by considering a monolingual corpus as a parallel corpus: a sentence is considered parallel to the paragraph that contains it. Then, IBM Model 1 [Brown et al. 1990] is used to determine this probability. In fact, the relationship $t(q_i|w_j)$ estimated in this way is a kind of co-occurrence relationship. Despite the limitation of this estimation method, the translation model itself can be suitable one in which to integrate other types of term relationships. In Cao et al. [2005], this translation model is extended successfully to integrate both co-occurrence and WordNet relationships. In this article, we propose to further generalize this framework to integrate more inference operations.

Before developing our method, let us first review some relevant logical approaches to inferential IR proposed in previous research projects.

3. LOGICAL APPROACHES

Since the 1980s, many researchers have believed that further improvements in IR will have to integrate logical inference. van Rijsbergen [1986] proposes modeling the relevance of a document D to a query Q as a logical implication, $D \rightarrow Q$. If a query is implied by a document, then the latter is considered to be relevant. To deal with uncertainty in this implication, a logical uncertainty principle for IR is proposed. The uncertainty $P(D \rightarrow Q)$ is determined by the amount of information that one has to add into the document to satisfy the query. The more information that is added to a document, the more uncertainty is attached to the relevance relation. This uncertainty principle is general and does not suggest a particular implementation. Since then, several studies have tried to implement it in different ways.

Crestani and van Rijsbergen [1995] propose to use logical imaging as a method to evaluate the probability of a counterfactual conditional, denoted by $>$ [Lewis 1973]. To evaluate the probability $P(D > Q)$, logical imaging tries to transfer the probabilities of terms to their closest neighbors appearing in D, creating a new probability distribution $P_D(.)$. Then the probability $P(D > Q)$ is evaluated as $P_D(Q)$. However, the experiments have not shown that the

approach can outperform the classic IR approaches. Nie and Lepage [1998] developed a general framework based on conditional logic to take into account the retrieval context: given a retrieval context, a document under consideration is considered to contain some new information. This later is assimilated into the context, making a new context that simulates the situation where the information described by the document is acquired. Then the query is examined with respect to the new context to see if it is satisfied. This model is capable of considering contextual factors such as the information already known to the user. However, the proposed model is difficult to implement in practice.

Nonmonotonic reasoning in IR has been examined in several studies. In particular, Bruza and Huibers [1996], Bruza et al. [2000], and Wong et al. [2001] consider *aboutness* as relevance, and they further propose an axiomatic system to formalize aboutness, which may be nonmonotonic. The logical properties of aboutness can be used to compare different models or systems, but they do not directly suggest an efficient way to implement an inferential IR system. Huibers et al. [1996] formulate inference in IR as information flow, that is, one tries to determine if a situation described in a document can entail that required by a query. To deal with uncertainty, Dempster-Shafer theory is used. Lau et al. [2004] and Lasado and Bareiro [2001] use belief revision to deal with nonmonotonic inference: a retrieval situation (including the user's knowledge or belief) is represented by a belief set which can contain ambiguity. Once a query is submitted, the belief set is revised in such a way that the knowledge that is the most coherent with the query is kept (see Lau et al. [2004] for details of the definition of coherence). Then documents are compared with the revised belief set to determine their degree of relevance.

In the studies presented on logical IR, it was generally assumed that classical logic is inappropriate for IR because of the utilization of the material implication $\supset$. This inappropriateness has been analyzed in several investigations [Nie and Lepage 1998, van Rijsbergen 1986, Bruza and Huibers 1996, Lalmas and Bruza 1998]. Two main types of inappropriateness have been pointed out.

(1) Classical logic is unable to do nonmonotonic reasoning, whereas inference in IR is nonmonotonic. For example, while a document about ski can be relevant to winter sport, a document containing both water and ski can mean waterskiing and becomes irrelevant to winter sport. In logic terms, we have $ski \rightarrow winter\ sport$, but not $ski \wedge water \rightarrow winter\ sport$. This is an example of nonmonotonic inference in IR. Classical logic is unable to account for it.
(2) Classical logic cannot deal with uncertainty which is inherent in IR. In classical logic, an implication $A \supset B$ can only be true or false. No probability or uncertainty measure is associated with it. However, in IR, not only do we need to determine if $D \supset Q$ holds, but we also have to assign a measure of its uncertainty so that document can be ranked.

A proper treatment for nonmonotonic reasoning in IR is a difficult enterprise. Until now, no adequate model has been able to deal with this problem and be implemented efficiently. On the other hand, the consideration of uncertainty in inference is mandatory in any attempt with inferential IR. In this situation,

we believe that it is useful to consider first the basic inference corresponding to that in the classical logic. Although this latter is unable to account for all the inference phenomena in IR especially nonmonotonicity, it does capture the central part of the inference in IR. In the current state of IR where we cannot describe document contents and information needs precisely in strict logic expressions, it is reasonable to first integrate the basic part of inference in IR in an efficient way. This is the approach we take in this article. Therefore, in the remaining part of the article, we will not deal with nonmonotonicity of inference in IR and restrict ourselves to the classical form of inference.

As formulated in Nie [2003], the essential part of inference in IR corresponds to the following transitivity of implication:

$$A \rightarrow B \wedge B \rightarrow C \Rightarrow A \rightarrow C.$$

If we interpret $\rightarrow$ as relevance as proposed by van Rijsbergen [1986], then we can consider the following cases of relevance between a document D and a query Q:

$$D \rightarrow Q' \wedge Q' \rightarrow Q \Rightarrow D \rightarrow Q$$
$$D \rightarrow D' \wedge D' \rightarrow Q \Rightarrow D \rightarrow Q.$$

These formulas can be read as follows. To determine that D is relevant to Q,

(1) we can determine if there is a new form of query Q' such that Q' implies the original query Q, and that the new query Q' is satisfied (implied) by the original document;
(2) or we can determine if there is a new form of document D' such that D' is implied by the original document D and that D' satisfies Q.

Query expansion can be considered as a particular case of the first formula, that is, the new form of query Q' is the one expanded from Q by adding new terms that are related to Q. This new query is thought to imply Q, that is, $Q' \rightarrow Q$. If a document matches the new query, then it is assumed to match the original query and is retrieved. This transitivity property can be applied multiple times. However, if the implication relation is associated with uncertainty (as it is in our case), then the more we have to apply transitivity to satisfy $D \rightarrow Q$, the more $D \rightarrow Q$ becomes uncertain. This uncertainty aspect will be taken into account later in our model. Similarly, the second formula can be called an approach using document expansion.

Unfortunately, it is difficult to propose a general framework to implement the previous approaches directly. The main problem is that there may be many related forms of document and query, and they are not independent. If all the dependences have to be taken into account, we will obtain a very inefficient system. Therefore, simplifications have to be made.

In this article, we propose to implement inference in the LM framework, making the same simplification assumptions. There are several reasons to choose LM as our implementation tool.

—LM is robust to noise. As IR always deals with noisy and incomplete document and query expressions, the capability of reasoning in a noisy context

is important. This sharply contrasts LM to the classical rule-based inference in AI.

—LM makes a series of simplification assumptions in order to implement it efficiently. Although these assumptions are not true in reality, the success in experiments shows that they are a good compromise between efficiency and the capability of reasoning.

—LM is a framework flexible enough to integrate different types of relations. In addition to the co-occurrence relations, there are also manually-identified relations stored in thesauri. All these relations can be integrated in LM in the same way.

In the next section, we will describe in detail our inferential language models.

4. GENERAL INFERENTIAL LANGUAGE MODELS

Our goal in this section is to define LMs with some capability of logical inference, while keeping them efficient. We formulate the degree of certainty of relevance $P(D \rightarrow Q)$ as $P(Q|D)$. As in other LM-based approaches, we also assume here that query terms are independent. Then we have the same general formulation of the generative model and a model based on KL-divergence.

1. Generative model

$$\begin{aligned} P(D \rightarrow Q) &= P(Q|D) \\ &= \prod_{i=1}^{n} P(q_i|D). \end{aligned}$$

2. KL-divergence:

$$\begin{aligned} R(D, Q) &= \log P(D \rightarrow Q) \\ &\propto \sum_{q_i \in V} P(q_i|Q) \log P(q_i|D). \end{aligned}$$

Notice that we no longer assume an ML estimation for query model $P(.|Q)$ in KL-divergence, and we consider all the words in vocabulary (V) in this general form.

The problem now is to determine good document and query language models, that is, $P(.|D)$ and $P(.|Q)$. In the previous LM approaches, these models are built using the following two distributions (a) the term distribution in the document or query, and (b) the term distribution in the whole document collection. As we mentioned earlier, it is possible that after a classical smoothing process, related and unrelated terms are attributed with similar probabilities. Our intended smoothing method tries to expand the original document or query models so that the related terms have higher probabilities. This is a new type of semantic smoothing that exploits the relationships between terms.

In what follows, we will first formulate two basic approaches that integrate this idea. Then some further extensions will be considered.

4.1 Inference as Document Expansion in Generative Model

Inference in documents tries to infer implied information from a document. This means building a new document model which integrates related terms.

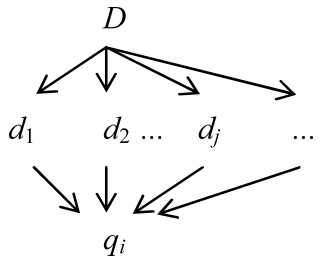

Fig. 1. Illustration of inference of a single query term.

The following logical inference describes what we want:

$$D \rightarrow d_j \wedge d_j \rightarrow q_i \Rightarrow D \rightarrow q_i.$$

That is, to determine if (and to what degree) a query term q_i is true in a document ($D \rightarrow q_i$), we observe whether this term is implied by another term d_j in the document. The appearance of a related term in the document means that the query term is also satisfied by the document to some degree.

The first part $D \rightarrow d_j$ of the previous formula can be modeled by a classical document LM, $P(d_j|D)$. The key to implementing inference is to add the second component, $d_j \rightarrow q_i$, that represents the relationships between terms. We will use $P_R(q_i|d_j)$ to represent $P(d_j \rightarrow q_i)$, the probability that q_i is implied by d_j through term relationships. These relationships will be determined either through term co-occurrences or a manual thesaurus (Wordnet). We will describe this in detail later. Now let us assume that we have the function $P_R(q_i|d_j)$. Then the intended logical inference corresponds to the following formulation in language modeling framework, which creates a new LM $P_R(.|D)$:

$$P_R(q_i|D) = \sum_{d_j \in D} P(d_j|D) P_R(q_i|d_j).$$

This formulation is similar to the translation model [Berger and Lafferty 1999]. Implicitly, we also assumed that the relations between a document term and the query term are independent as in Berger and Lafferty [1999], making it possible to sum up over all the document terms. Again, this simplification is made to make the model tractable.

This approach can be easily reformulated in terms of logic:

$$P_R(q_i|D) = P(D \rightarrow q_i) = \sum_{d_j \in D} P(D \rightarrow d_j) P(d_j \rightarrow q_i).$$

Figure 1 illustrates the corresponding inference process on a document.

Notice that the inference of the query term via another term in the figure is one way to prove that the query term is satisfied by the document. Another possible way is the direct appearance of the query term in the document. This latter case can be included in the first one by assuming that $q_i \rightarrow q_i$ is a valid relation. Another way to consider the self implication is to smooth the model $P_R(.|D)$ with a traditional unigram model $P(.|D)$ which can be considered as a particular case of $P_R(.|D)$ defined previously, but with $P(q_i \rightarrow q_i) = 1$ and $P(q_i \rightarrow q_j) = 0$ for $q_i \neq q_j$. By fixing an appropriate smoothing parameter,

we can arrive at a similar result to the first approach. Formulated in terms of language modeling, the smoothed model P_E is defined as follows:

$$P_E(q_i|D) = \lambda P_R(q_i|D) + (1 - \lambda)P(q_i|D),$$

where $P(q_i|D)$ is a classical document language model (without inference) and λ is a smoothing parameter. The model $P_E(q_i|D)$is the resulting expanded document model.

By this smoothing, we consider indeed that there are two ways for a document to satisfy a query term: either the query term appears in the document, or a related term appears in it. The latter case is exactly the desired inference process.

4.2 Inference as Query Expansion

Conceptually, similar inference can also be made on query. One would believe that it is easy to directly implement inference as query expansion in a symmetric way to document expansion. In fact, this is difficult with a generative model because a query is not an explicit unit in it, but instead is decomposed into separate words. For query expansion to make sense, there has to be a specific model for query which is the case of the approach using KL-divergence.

In the KL-divergence approach, we build both a document model and query model, and measure the difference (divergence) between them. As we mentioned, in the previous formulation, usually only the document model is smoothed with the collection model and the query model uses maximum likelihood estimation. One can observe that the difference measured in this way is partial since a query usually only contains a few words. With a smoothed query model, the difference between document and query could be better measured. In some recent work [Zhai and Lafferty 2001b; Lavrenko and Croft 2001, Kurland et al. 2005], the query model is also smoothed with a set of feedback documents in different ways. Pseudorelevance feedback is known to be an effective way to complete the partial expression of a query in order to find more relevant documents. However, smoothing using feedback documents still relies only on a redistribution of probability without considering term relationships. Our query smoothing is based on term relationships. The expanded query model $P_E(q_i|Q)$ can be formulated as follows:

$$\begin{aligned} P_E(q_i|Q) &= \lambda P_R(q_i|Q) + (1 - \lambda)P_{ML}(q_i|Q) \\ P_R(q_i|Q) &= \sum_{q_j \in V} P_{ML}(q_j|Q) * P_R(q_i|q_j) = \sum_{q_j \in Q} P_{ML}(q_j|Q) * P_R(q_i|q_j), \end{aligned} \tag{3}$$

where V is the vocabulary.

However, from the practical point of view, this query model smoothing has a serious problem. If the term relationship $P_R(q_i|q_j)$ is extracted from data using a statistical method, then very likely a given query term q_j would have a relationship with many terms q_i in the vocabulary. This would result in a very large set of query terms with nonzero probability and inefficient query evaluation.

To solve this problem, we observe that, in general, a given term will have a relatively strong relationship with a small number of terms and a very weak relationship with the other majority of terms. The latter can be considered

as the noisy part of the relationship and can be ignored. From this point of view, we can limit the query expansion process only to the strongest part of the relationship, for example, by limiting the number of expansion terms or by selecting the terms with a sufficient degree of relationship with an original query term. Let E be the set of the most strongly related terms selected, and let Q_E be the resulting expanded query. Then we have $P_E(q_i|D) = 0$ for all $q_i \notin Q_E$. A score according to KL-divergence formula can be simplified to the following:

$$R(D, Q) = \sum_{q_i \in Q_E} P_E(q_i|Q) \log P(q_i|D).$$

It is easy to see that the this query expansion approach is not strictly equivalent to the earlier document expansion approach because the equivalence can only be established when we use ML estimation for the query model. However, there is still a close relationship as we will show in the following.

Indeed, the query smoothing process can also be understood as a traditional process of query expansion which tries to create a new query Q_E by adding new related terms. The new probability function $P_E(.|Q)$ can also be considered as a ML estimation for the expanded query Q_E if we assume the following word counts in Q_E:

$$c(q_i; Q_E) = \begin{cases} \lambda c(q_i; Q) + (1 - \lambda) P_R(q_i|Q) \times |Q|, & \text{if } q_i \text{ is in the original query;} \\ (1 - \lambda) P_R(q_i|Q) \times |Q|, & \text{if } q_i \text{ is a newly added term.} \end{cases}$$

We can easily check that $P_E(q_i|Q) = P_{ML}(q_i|Q_E)$. Then the resulting score function can be rewritten as follows:

$$\begin{aligned} R(D, Q) &= \sum_{q_i \in Q_E} P_E(q_i|Q) \log P(q_i|D) = \sum_{q_i \in Q_E} P_{ML}(q_i|Q_E) \log P(q_i|D) \\ &\propto \sum_{q_i \in Q_E} c(q_i; Q_E) \log P(q_i|D) = \log \prod_{q_i \in Q_E} P(q_i|D)^{c(q_i; Q_E)} = \log P(Q_E|D). \end{aligned}$$

We see that the corresponding query evaluation process is still equivalent to a generative model with the expanded query. This approach can be formulated as a particular case of $P(D \to Q_E)P(Q_E \to Q)$, with $P(Q_E \to Q) = 1$ for the above Q_E.

Both document and query expansion approaches implement the basic ideas of inference in IR. They can be enhanced in several respects. We will now examine two such aspects, namely, integration of specificity in inference and multistep inference.

4.3 Specificity and Exhaustivity

The above inference processes can go through an inference path via some intermediate terms w. The higher $P(D \to w)P(w \to q_i)$, the more the word w will play an important role in the retrieval process. From a logical point of view, this process is reasonable. However, it does not distinguish between a general word and a specific word as the intermediate term w. Imagine that to infer the word computer from a document about algorithms, we have two possible intermediate terms programming and part. Suppose that these two

words appear in that document. Then a possible path to infer computer goes through part with probability $P(D \rightarrow part)P(part \rightarrow computer)$. If term relationships are defined manually, it is unlikely that part is related to computer, and $P(part \rightarrow computer) > 0$. So this path is invalidated. However, if term relationships are extracted from documents based on co-occurrences, then it is possible that $P(part \rightarrow computer) > 0$ because the frequent word part may co-occur often with many other words. As a consequence, the path $P(D \rightarrow part)P(part \rightarrow computer)$ is considered as a possible one to infer computer from the document. Obviously, this is not a desirable inference path. What we want is to infer computer through another term, namely, programming, in the path $P(D \rightarrow programming)P(programming \rightarrow computer)$. The previous inference processes cannot distinguish between these two paths. To make things worse, in general, a more general term w (such as part) usually has a higher$P(D \rightarrow w)$ than a more specific term for many documents. Even if $P(w \rightarrow q_i)$ is lower for a general w, when both probabilities are combined, general terms can be selected to the detriment of more specific terms. In the case of query expansion, selecting more general expansion terms often leads to noise expansion terms.

What is missing in our formulation so far is the consideration of specificity. In fact, logical modeling often tries to formulate one-directional implication, that is, $w \rightarrow q_i$ or $D \rightarrow Q$ for the complete document and query, as proposed in van Rijsbergen [1986]. When no uncertainty is involved, this makes perfect sense. However, when uncertainty occurs, this implication is not all we want. When we select a perfect document for a query, we want to ensure that, on the one hand, the document is about all the aspects of the query, and, on the other hand, the document concentrates on the topic of the query and does not talk about many other topics. The first factor is called *exhaustivity* of the document for the query, and the second one *specificity*. They are represented respectively by the following logical expressions $D \rightarrow Q$ and $Q \rightarrow D$ in Nie and Lepage [1998]. The same concepts also apply to a single query term $D \rightarrow q$ and $q \rightarrow D$.

The problem we observed in one-directional implication is closely related to the lack of specificity. In order to take this into account, we propose to use $w \leftrightarrow q_i$ (which denotes logical equivalence) instead of $w \rightarrow q_i$. In this way, the logic inference becomes:

$$D \rightarrow w \wedge w \leftrightarrow q_i \Rightarrow D \rightarrow q_i.$$

This is a stricter reasoning schema than the earlier implication transitivity. This formulation is more reasonable in IR. The addition of the reverse implication $q_i \rightarrow w$ allows us to avoid selecting too general expansion terms. In our model, we propose to use the following formula to evaluate $P(w \leftrightarrow q_i)$:

$$P(w \leftrightarrow q_i) = \alpha[P(w \rightarrow q_i)]^{\gamma}[P(q_i \rightarrow w)]^{(1-\gamma)} = \alpha P(q_i|w)^{\gamma} P(w|q_i)^{(1-\gamma)},$$

where $0 < \gamma < 1$ is a factor that determines the relative importance of each implication, and α is a normalization factor that ensures $P(q_i \leftrightarrow w)$ is a probability function, that is, $\sum_{q_i} P(w \leftrightarrow q_i) = 1$. This formula corresponds to an average of the implications in both directions. As we will see in our experiments, the consideration of implications in both directions produces better results.

4.4 Multistep Inference Using Markov Chain

The inference processes described so far use only one-step inference. However, inference should not be limited to one step: if "algorithm" is related to "programming" and "programming" to "computer", then "algorithm" is also related to "computer". So, a more complete inference can be achieved by allowing multistep inference. This multistep inference process can be formulated as a Markov chain or random walk as follows.

Let E be a set of terms that are strongly related to the original query as before, and $w \in E$ be a possible expansion term. To use a Markov chain or random walk, w is viewed as a state, $P_0(w|Q)$ is the initial distribution, which can also be viewed as the prior probability of term w. One way to define it is to use $P_E(w|Q)$ defined in Equation (3). We will see in the next section that we can also incorporate feedback information in defining it. $P_R(w|w')$ is considered as the probability of state transition from w' to w. Then the query model using the transition can be defined as:

$$P_t(w|Q) = \sum_{w' \in E} P(w|w') P_{t-1}(w'|Q),$$

where $P_t(w|Q)$ is the query model after t-th updating. To ensure this Markov chain has a unique stationary distribution, we define the probability as:

$$P(w|w') = \gamma P_0(w|Q) + (1-\gamma) P_R(w|w'). \tag{4}$$

With up to T transitions, the resulting probability of w in query Q is

$$P_T(w|Q) = \gamma \sum_{t=0}^{T} (1-\gamma)^t P_t(w|Q).$$

The Markov chain just defined has a unique stationary distribution and it can be calculated as:

$$P_M(w|Q) = \lim_{T \to +\infty} P_T(w|Q) = \gamma \sum_{t=0}^{\infty} (1-\gamma)^t P_t(w|Q).$$

This function $P_M(w|Q)$ is used as the final query model.

The previous inference process integrates up to T steps of inference. Therefore, the inferential power is increased. Of course, the longer the inferential process is, the more uncertain the resulting terms are related to the original query. This is characterized by $(1-\gamma)^t$ in the preceding formula. Although $T \to +\infty$ is used in the equation, we observe in our experiments that a limited number of transitions (about 10) are sufficient to make $P_M(w|Q)$ converge.

5. IMPLEMENTATION

5.1 Basic Language Model

In all cases, we need a basic language model for a document—smoothed unigram model. In our implementation, we use the following model that interpolates ML

estimation with an absolute discount [Zhai and Lafferty 2001a]:

$$P_{abs}(w_i|D) = \frac{\max(c(w_i;D) - \delta, 0)}{|D|} + \frac{\delta|D|_u}{|D|} P_{ML}(w_i|C), \tag{5}$$

where δ is the discount factor (which is set at 0.7 as suggested in Zhai and Lafferty [2001a]), $|D|$ is the length of the document, $|D|_u$ is the count of unique terms in the document, and $P_{ML}(w_i|C)$ is the maximum likelihood probability of the word in the collection C. This smoothing method is chosen among a set of other smoothing methods (such as Jelinek-Mercer smoothing and Dirichlet smoothing) because it resulted in the most stable performance in our experiments.

5.2 Term Relationships

To implement the models described in the previous section, we first need to determine term relationships $P_R(w_i|w_j)$. Several methods are possible [Cao et al. 2005]:

—One can derive term relationships by analyzing term co-occurrences. This is one of the common ways used in IR. The assumption is that the more two terms co-occur in the same windows of some size (empirically set at 7 in our case), the more they are considered to be related. Then a probability function $P(w_i|w_j)$ can be defined as follows:

$$P_{CO}(w_i|w_j) = \frac{c(w_i, w_j)}{\sum_{w_k \in V} c(w_k, w_j)},$$

where $c(w_i, w_j)$ is the count of co-occurrences of w_i and w_j in the document collection.

—Several linguistic resources have been created manually that contain relationships between terms. WordNet [Miller 1990] is an example. Such a resource can also be used to define another function: $P_{WN}(w_i|w_j) > 0$, if there is a relationship between w_j and w_i in the linguistic resource. A problem arises when such a manually-prepared resource is used; usually no numerical value is available to define $P(w_i|w_j)$. In order to arrive at a definition of the required probability function, we combine the manual resource with co-occurrences according to the following principle. The probability of a term relation is nonzero only if the terms have a relationship in the manual resource; the more these terms co-occur in the same windows, the stronger their relationship is. Then we can arrive at the following definition of another term relationship $P_{WN}(w_i|w_j)$ using WordNet and co-occurrences:

$$P_{WN}(w_i|w_j) = \frac{c_{WN}(w_i, w_j)}{\sum_{w_k \in V} c_{WN}(w_k, w_j)},$$

where $c_{WN}(w_i, w_j)$ is the count of co-occurrences of two words with a relationship in WordNet. The window used for WordNet relationships are different from the one for previous co-occurrence relations: We consider a paragraph as a window.

5.3 Combination of Different Types of Term Relationships

Term relationships from different resources have different characteristics. Those stored in a manual thesaurus such as WordNet are manually validated, but they are often ambiguous and incomplete. The relationships extracted from document collections are strongly related to the area of the document collection, and the coverage may be relatively good. However, much noise (false relationships) will also be extracted. Therefore, a good approach is to combine both types of relationship. Such a combination has been used in Mandala et al. [1998]. Here, we use smoothing for this combination. Then in a document expansion approach, the expanded part of the new document model with both WordNet and co-occurrence relationships is as follows:

$$\begin{aligned} &P_R(q_i|d_j) = \lambda_{CO} P_{CO}(q_i|d_j) + (1-\lambda_{CO}) P_{WN}(q_i|d_j) \\ &P_R(q_i|D) = \sum_{d_j \in D} P_R(q_i|d_j) P(d_j|D) \\ &= \lambda_{CO} \sum_{d_j \in D} P_{CO}(q_i|d_j) P(d_j|D) + (1-\lambda_{CO}) \sum_{d_j \in D} P_{WN}(q_i|d_j) P(d_j|D). \end{aligned} \tag{6}$$

Defining $P_{CO}(q_i|D) = \sum_{d_j \in D} P_{CO}(q_i|d_j) P(d_j|D)$ and $P_{WN}(q_i|D) = \sum_{d_j \in D} P_{WN}(q_i|d_j) P(d_j|D)$, we have :

$$P_R(q_i|Q) = \lambda_{CO} P_{CO}(q_i|D) + (1-\lambda_{CO}) P_{WN}(q_i|D).$$

Finally, the preceding model is smoothed with the classical (noninferential) document unigram model $P_U(q_i|D)$ to obtain the final model:

$$P_E(q_i|D) = \lambda_R P_R(q_i|D) + (1-\lambda_R) P_U(q_i|D).$$

The final document model is illustrated in Figure 2. In this figure, we can see that the probability of a query term q_i in a document is determined through three paths: by the classical unigram model, or by one of the relation models. All the paths are assumed to be independent.

The parameters λ_R and λ_{CO} can be tuned by EM that tries to maximize the likelihood of the query Q by the whole collection (N documents). Let θ be the set of all the parameters of the model, and $\theta_q = [\lambda_R, \lambda_{CO}]$ be a subset to be tuned. Then the best θ_q is such that:

$$\begin{aligned} \theta_q^* &= \arg\max_{\theta_q} \log P(Q|\theta) \\ &= \arg\max_{\theta_q} \log \sum_{i=1}^{N} \pi_i \prod_{j=1}^{n} [(1-\lambda_R) P_U(q_j|D_i) + \\ &\qquad \lambda_R((1-\lambda_{CO}) P_{WN}(q_j|D_i) + \lambda_{CO} P_{CO}(q_j|D_i))], \end{aligned}$$

where $\{\pi_i\}_{i=1}^N$ are a set of parameters that characterize the closeness of the feedback documents to the query. They are not fixed, but tuned during the EM process. This allows us to allocate higher weight to documents that generate the query well and presumably are also more likely to be relevant.

However, the top N retrieved documents also contain nonrelevant documents. To account for the noise, we further assume that these documents are

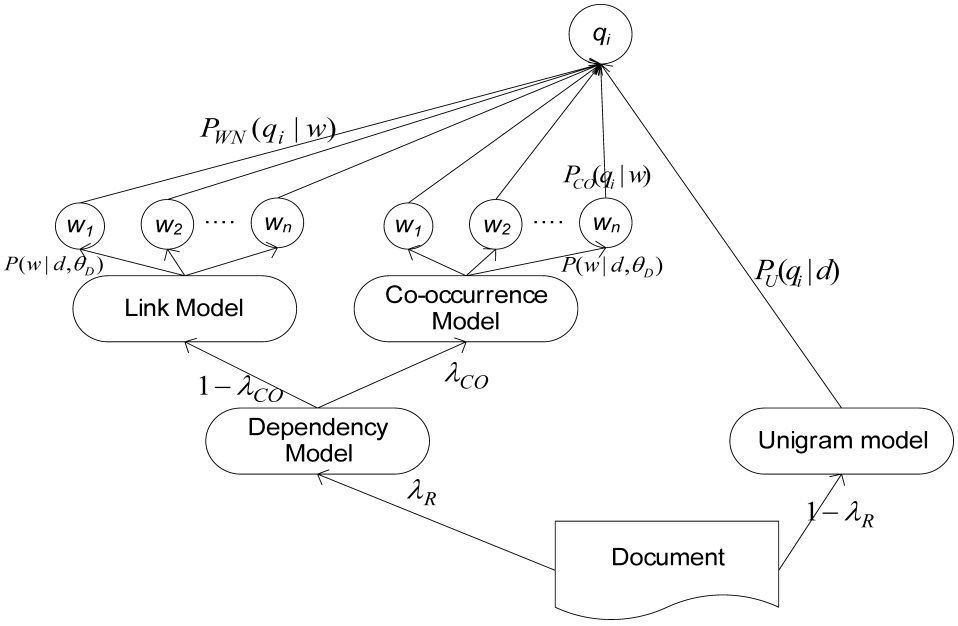

Fig. 2. Illustration of inference of a query term from a document.

generated from two sources, relevant documents and a noise source which is approximated by the collection C. Then the previous equation can be rewritten as follows:

$$\theta_q^* = \arg\max_{\theta_q} \log \left\{ \begin{array}{l} (1-\mu)\sum_{i=1}^{N} \pi_i \prod_{j=1}^{n} [(1-\lambda_R)P_U(q_j|D_i) \\ \quad + \lambda_R((1-\lambda_{CO})P_{WN}(q_j|D_i) + \lambda_{CO}P_{CO}(q_j|D_i))] \\ +\mu \prod_{j=1}^{n} [(1-\lambda_R)P_U(q_j|C) \\ \quad + \lambda_R((1-\lambda_{CO})P_{WN}(q_j|C) + \lambda_{CO}P_{CO}(q_j|C)))] \end{array} \right\},$$

where μ is the weight of the noise. μ is set at a nonzero value (0.3 in our experiments), otherwise it would become close to zero because in this way, the documents would have higher likelihood. In fact, the role of α is to add some robustness counteract to the noise of the training data. In EM, the corresponding update formulas are as follows (we do not provide details for their derivation here):

$$\pi_i^{(r+1)} = \frac{\pi_i^{(r)} \prod_{j=1}^{n} [(1-\lambda_R^{(r)})P_U(q_j|D_i) + \lambda_R(1-\lambda_{CO}^{(r)})P_{WN}(q_j|D_i) + \lambda_R\lambda_{CO}^{(r)}P_{CO}(q_j|D_i)]}{\sum_{i=1}^{N} \pi_i^{(r)} \prod_{j=1}^{n} [(1-\lambda_R^{(r)})P_U(q_j|D_i) + \lambda_R(1-\lambda_{CO}^{(r)})P_{WN}(q_j|D_i) + \lambda_R\lambda_{CO}^{(r)}P_{CO}(q_j|D_i)]},$$

and

$$\lambda_R^{(r+1)} = \frac{1}{n} \frac{\begin{array}{c}(1-\mu)\sum_{i=1}^{N} \pi_i^{(r)}[\lambda_R^{(r)}(1-\lambda_{CO}^{(r)})P_{WN}(q_j|D_i) + \lambda_R^{(r)}\lambda_{CO}^{(r)}P_{CO}(q_j|D_i)] + \\ \mu\lambda_R^{(r)}[(1-\lambda_{CO}^{(r)})P_{WN}(q_j|C) + \lambda_R^{(r)}\lambda_{CO}^{(r)}P_{CO}(q_j|C)]\end{array}}{\begin{array}{l}\{(1-\mu)\sum_{i=1}^{N} \pi_i^{(r)}[(1-\lambda_R^{(r)})P_U(q_j|D_i) + \lambda_R^{(r)}(1-\lambda_{CO}^{(r)})P_{WN}(q_j|D_i) + \lambda_R^{(r)}\lambda_{CO}^{(r)}P_{CO}(q_j|D_i)] + \\ \mu[(1-\lambda_R^{(r)})P_U(q_j|C) + \lambda_R^{(r)}(1-\lambda_{CO}^{(r)})P_{WN}(q_j|C) + \lambda_R^{(r)}\lambda_{CO}^{(r)}P_{CO}(q_j|C)]\}\end{array}}.$$

$$\lambda_{CO}^{(r+1)} = \frac{1}{n} \frac{(1-\mu)\sum_{i=1}^{N} \pi_i^{(r)} \lambda_R^{(r)} \lambda_{CO}^{(r)} P_{CO}(q_j|D_i) + \mu \lambda_R^{(r)} \lambda_{CO}^{(r)} P_{CO}(q_j|C)}{\{(1-\mu)\sum_{i=1}^{N} \pi_i^{(r)} [(1-\lambda_R^{(r)}) P_U(q_j|D_i) + \lambda_R^{(r)}(1-\lambda_{CO}^{(r)}) P_{WN}(q_j|D_i) + \lambda_R^{(r)} \lambda_{CO}^{(r)} P_{CO}(q_j|D_i)] + \mu[(1-\lambda_R^{(r)}) P_U(q_j|C) + \lambda_R^{(r)}(1-\lambda_{CO}^{(r)}) P_{WN}(q_j|C) + \lambda_R^{(r)} \lambda_{CO}^{(r)} P_{CO}(q_j|C)]\}}.$$

5.4 Improving Query Expression Using Feedback

A common problem in IR is that queries are usually short, 2–3 words in general for queries sent to search engines on the Web. Such a short description of information cannot be precise. Even in the case of a longer query as in TREC, we still cannot expect a perfect description of the information need because there are always missing aspects, and there may be alternative descriptions of it. One way to enrich the query expression is to use feedback documents. We do a first retrieval using the original query; the top n retrieved documents (feedback documents) are assumed to be relevant, and they are used to add new terms into the query or to define a new query model.

Let F be the set of feedback documents. A new query model can be created by combining the initial model with the feedback model as follows:

$$P_F(w|Q) = \lambda P_{ML}(w|Q) + (1-\lambda) P(w|F), \tag{7}$$

where $P_{ML}(w|Q)$ is the *ML* estimation probability of w in Q, $P(w|F)$ is the model created from document set F. It is important to use $P_{ML}(w|Q)$ in the above equation in order to prevent the query from drifting from the original query. We have a number of approaches to estimate $P(w|F)$, for example, ML estimation or with a relevance model Lavrenko and Croft [2001]. Here, we use a mixed model presented in Zhai and Lafferty [2001b]. In this model, we optimize $P(w|F)$ in order to maximize the likelihood of feedback documents. Let θ be the parameters of a possible model and θ^* be the optimal one. We have:

$$\begin{aligned}\theta^* &= \arg\max_{\theta} P(F|\theta) = \arg\max_{\theta} \prod_{D \in F} \prod_{w \in D} P(w|\theta)^{c(w;D)} \\ &= \arg\max_{\theta} \sum_{D \in F} \sum_{w \in D} c(w;D) \log P(w|\theta),\end{aligned}$$

where $c(w;D)$ is the count of w in D.

As in Section 5.3, this model will cause overfitting, and θ^* will correspond to the ML model. So, we also add a noise source, which is simulated by the collection model as follows:

$$\theta^* = \arg\max_{\theta} \log P(F|\theta) = \arg\max_{\theta} \sum_{D \in F} \sum_{w \in D} c(w;D) \log((1-\mu) P(w|\theta) + \mu P(w|C)).$$

The EM algorithm is used again to obtain θ^*, with the following update formulas:

$$t^{(n)}(w) = \frac{(1-\mu)P^{(n)}(w|\theta)}{(1-\mu)P^{(n)}(w|\theta) + \mu P(w|C)}$$

$$P^{(n+1)}(w|\theta) = \frac{\sum_{D \in F} c(w;D)t^{(n)}(w)}{\sum_{D \in F}\sum_{w \in D} c(w;D)t^{(n)}(w)}.$$

The final value of $P(w|\theta)$ is used as $P(w|F)$. In practice, the values of t and p converge quickly. In our experiments, we only need to iterate about 10 times.

The model $P_F(w|Q)$ determined by Equation (7) can be used in two ways.

—It can be used as the final query model and to calculate a score based on KL-divergence. To do this, one also has to limit the number of expansion words added into the query, otherwise the query evaluation process would be inefficient. In our case, we choose 80 expansion terms. This corresponds to the model MixM that we test in Section 6.3.

—It can be used as the initial probability $P_0(w|Q)$ in Equation (4). The Markov chain (random walk) model then performs multistep inference to derive new expansion terms.

5.5 Extracting Term Relationships from Feedback Documents

In Equation (7), a new LM is created for the feedback documents F. However, it is still limited to a term distribution. As the feedback documents usually correspond better to the query than the documents in the collection in general, it is also possible to extract term relationships from F. In doing so, the extracted term relationships may become more relevant to the particular area of the query. The experiments in Xu and Croft [1996] show that the expansion terms extracted from feedback documents (called local context analysis) are better than those extracted from the whole collection (global context analysis). Following the same idea, once we identified a set (20) of feedback documents, we extracted co-occurrence relationships from these documents, and these relationships were combined with the co-occurrence relationships extracted from the whole collection. The following formula describes this approach:

$$P_{RF}(w|w') = \lambda_1 P_R(w|w') + (1-\lambda_1)P_F(w|w'), \tag{8}$$

where $P_R(w|w')$ is the relationship extracted from the whole collection, and $P_F(w|w')$ is the relationship extracted from the feedback documents. For English, both of them integrate co-occurrence relationships and WordNet relationships as before. For Chinese, they are co-occurrence relationships only. The parameter λ_1 is set between 0.8–0.9. The effectiveness is quite stable with respect to this parameter so we did not tune it through EM (although this is possible).

Table I. Statistics of Data Set

Coll.	Description	Size (MB)	# Doc.	Vocab. Size	Avg Doc Len	Query	Avg Qry Length
AP	*Associate Press* (1988–90), Disks 2&3	729	242,918	245,748	244	TREC topics 51–100	13
WSJ	*Wall Street Journal* (1990–92), Disk 2	242	74,520	121,944	264	As AP	13
SJM	*San Jose Mercury News* (1991), Disk 3	287	90,257	146,512	217	As AP	13
CH1	People's Daily (91–93) & Xinhua Daily (94–95)	162	164,789	274,901	242	TREC topics CH1-28 (Title)	7.1
CH2	As CH1	As CH1	As CH1	As CH1	As CH1	As CH1 (Title + Desc)	18

$P_{RF}(w|w')$ is used as the new transition probability in Equation (4) for random walk. This is the model identified as RandM tested in Section 6.3.

6. EXPERIMENTS

In order to test the effectiveness of the models described in the last section, we carry out a series of experiments on four TREC collections—three of them are English collections and one is a Chinese collection. For the Chinese collection, we test with both short queries (title) and long queries (title and description). The utilization of the Chinese collection aims to show that the proposed method is language independent and can be used on different languages. Table I shows the statistical information of the test collections.

6.1 Data Preprocessing

All English documents have been preprocessed in a standard manner; terms were stemmed using the Porter stemmer and stopwords were removed. The queries are Topics 51–100 used in TREC. We used the title and description fields of the topics. These queries contain 13 words on average. The document set comes from the TREC disks 2 and 3.

For English, we also use WordNet as a source of term relationships. We use WordNet version 2.0. For each word in the vocabulary of the dataset, we extract its synonym, hypernym, and hyponym from WordNet and build a pool of related terms for it. The processing is done offline. As we do not explicitly consider compound terms, all the compound terms in WordNet are decomposed into their component words.

For Chinese, the entire dataset (including the documents and queries) was converted into GB2312 encoding. We carried out a dictionary-based word segmentation. The dictionary contains 137,613 Chinese words. This dictionary is compiled by UC Berkley, and it has been used in several TREC experiments on Chinese IR. According to Foo and Li [2004], we can obtain the best IR results when most segmented Chinese words have two characters. Therefore, we limit the length of the word to 3 characters when we segment Chinese texts and queries. The queries and documents are processed in the same way, and we do not filter out any stopword. Because there is not a counterpart to WordNet in Chinese, we only use co-occurrence relationships, that is, λ_{CO} in Equation (6) is set to 1.

Table II. Comparison between Unigram Model and Document Expansion Model

	Unigram Model		Document expansion		
Coll.	AvgP	Recall	AvgP	% change	Recall
AP	0.1925	3289/6101	0.2128	+10.54**	3523/6101
WSJ	0.2466	1659/2172	0.2597	+5.31*	1704/2172
SJM	0.2045	1417/2322	0.2142	+4.74	1572/2322
CH1	0.2644	1697/2182	0.2776	+4.99	1754/2182
CH2	0.3483	1918/2182	0.3571	+2.53*	1962/2182

* and ** indicate that the difference is statistically significant according to t-test at the level of $p\text{-}value < 0.05$ and $p\text{-}value < 0.01$.

6.2 Experiments on Document Expansion

For English, our document expansion model is that described in Section 5.3. In Table II, we show the results obtained for each test collection. The unigram model is the basic LM that implements Equation (6) using absolute discounting smoothing. The document expansion model implements Equation (7).

Table II compares the results of the unigram model and document expansion model. AvgP is the noninterpolated mean average precision. Recall shows the number of relevant documents that are retrieved among the top 1,000 results over all relevant documents.

As we can see, on all the test collections, the document expansion model outperforms the basic unigram model. In 3 of the 5 experiments, the improvements are statistically significant (at the level of p-value < 0.05 or p-value < 0.01). This series of experiments shows that inference implemented as document expansion can improve IR effectiveness on both English and Chinese documents.

6.3 Experiments on Query Expansion

We test the following three query expansion models.

QE. This is the basic query expansion model, which uses term relationships extracted from co-occurrences and from WordNet (for English). We use 80 expansion terms in this experiment. This model corresponds to Equation (6). The experiments aim to show the contribution of inference in query expansion based on term relationships.

MixM. This is the mixed model which is presented in Zhai and Lafferty [2001b] which uses feedback documents. In this experiment, the original query is first expanded using term relationships from the whole collection. The top 20 documents are used as feedback documents and 80 terms are smoothed with the original query model. These numbers are suggested by Zhai and Lafferty [2001b]. This model corresponds to Equation (7).

RandM. This is the random walk model which uses MixM as the initial distribution $P_0(w|Q)$ in Equation (4). The transition probability incorporates the co-occurrence relationships extracted both from the whole collection and from 20 feedback documents (Equation (8)) as well as WordNet for English. To compare the two models, we use the same values for the common parameters as MixM. The goal of this experiment is to show the impact of adding multistep inference on a model using feedback documents.

Table III. Comparison of Different Models for Query Expansion

	UM		QE			MixM			RandM		
Coll.	AvP.	Rec.	AvP.	%chg1	Rec.	AvP.	%chg1	Rec.	AvP.	%chg2	Rec.
AP	0.1925	3289/6101	0.1959	+1.76	3370/6101	0.2350	+22.07**	3700/6101	0.2543	+8.21*	4003/6101
WSJ	0.2466	1659/2172	0.2483	+0.68	1636/2172	0.2731	+10.75**	1730/2172	0.2842	+4.06	1786/2172
SJM	0.2045	1417/2322	0.2142	+4.74	1485/2322	0.2298	+12.37**	1526/2322	0.2535	+10.31*	1627/2322
CH1	0.2644	1697/2182	0.2646	+0.07	1699/2182	0.3261	+23.34**	1827/2182	0.3631	+11.53**	1919/2182
CH2	0.3483	1918/2182	0.3484	0	1919/2182	0.3841	+10.28**	1980/2182	0.3906	+1.69*	1993/2182

AvP. is the noninterpolation average precision. Rec. is the recall. chg is the improvement compared with UM. chg1 means the improvement over UM, and chg2 means the improvement over MixM.

All of the models incorporate specificity, that is, equivalence instead of one-directional implication. The parameter γ in Equation (7) is set at $\gamma = 0.3$ for all datasets except CH2 for which it is set at 0.8. We will see later that this is better than one-directional inference.

Table III shows the results of these models (where UM is the same unigram model as before).

We can see that the basic query expansion model (QE) only marginally outperforms the unigram model. For Chinese, in particular, virtually no improvement has been obtained. This result is not really surprising, and it is consistent with several studies on query expansion (e.g., Voorhees [1994]).

Comparing the results we obtained by document expansion, the poor result of query expansion suggests that the two approaches behave in different ways, although conceptually they are similar. In particular, query expansion seems to be more sensitive to noise introduced during expansion. Indeed, in document expansion, since there are many more original terms than in a query, even if an expansion introduces some noise terms, expansion terms may still converge globally to the related terms, and we can still obtain some reasonable expansion terms. For query expansion, however, such a convergence is much more difficult to obtain because of the small number of terms. This fact is directly related to the richness of context for expansion.

We show here the query 70 in the AP collection as an example. The original query is "Title: Surrogate Motherhood; Description: document will report judicial proceedings and opinions on contacts for surrogate motherhood". After query expansion, the strongest terms introduced into the query are sign, paper, account, state and so on. We can see that these terms are not strongly related to the query. Their introduction into the query brings noise.

In contrast, we can see in the column MixM that the utilization of a feedback model mixing with the original query model is highly effective. This result is similar to that of Zhai and Lafferty [2001b]. The addition of feedback documents clearly allows us to create a better query model.

What is interesting to observe is that, once feedback documents are used to enhance the query model, term relationships become more useful. This can be observed in the column RandM in which feedback documents are used in the following two ways: (1) to create a feedback unigram model as in MixM, and (2) to provide a subset of documents from which co-occurrence relationships are extracted. This provides us with more related expansion terms. For the same query just mentioned, the strongest (stemmed) terms suggested by new term

Table IV. One-Directional Inference v.s. Bidirectional Inference

	w→q		w↔q		
Collection	AvP.	Rec.	AvP.	%chg.	Rec.
AP	0.2533	3913/6101	0.2543	0.39	4003/6101
WSJ	0.2719	1774/2172	0.2842	4.53*	1786/2172
SJM	0.2433	1614/2322	0.2535	4.19*	1627/2322
CH1	0.3012	1726/2182	0.3631	20.55**	1919/2182
CH2	0.3859	1985/2182	0.3906	1.21*	1993/2182

Table V. Expanded Query with One-Directional Inference

insid 0.0438716	**trade** 0.0283044
discuss 0.0145507	**case** 0.0135872
document 0.0111333	market 0.00545457
stock 0.00388469	state 0.00371557
time 0.00362664	busi 0.00346713
exchang 0.003441	talk 0.00340819
thing 0.00335151	close 0.00308628
report 0.00305867	*part* 0.00299384
live 0.00290448	*person* 0.00269168
inform 0.00264513	*work* 0.00259297

relationships are: court, whitehead, year, suprem, gould, mother, case, and so on, which are more related to the query.

The improvement obtained with this model is not surprising. As the document collection contains documents on various topics, the term relationships extracted are also applicable to different areas. It is not possible for us to select only the relationships appropriate to the query (area). As a consequence, ambiguous words are expanded in all the possible areas, resulting in a very noisy query model. In contrast, when we extract term relationships from the feedback documents, filtering has been made during the first retrieval. It can be assumed that the relationships extracted from the feedback documents are more related to the query. This observation is similar to that on global and local context analysis in Xu and Croft [1996]. This provides another explanation as to why there is a large difference between QE and RandM in their improvements over UM and MixM, respectively.

6.4 The Effect of Integrating Specificity

The addition of reverse implication to account for specificity of expansion terms brings a notable improvement. Table IV shows the results obtained with one- and two-directional implications. We can see that the consideration of specificity is useful on all the collections. The increase of effectiveness on CH1 is the most important.

We observe that in general, when specificity is not considered, many queries are expanded with terms such as time, part. Table V contains the 20 strongest terms after expanding the query on "insider trading, document discusses an insider-trading case" (Query 55). The original terms are in bold. Table VI contains the strongest terms when we also use specificity in the query expansion.

Table VI. Expanded Query with Bidirectional Inference

insid 0.044366	**trade** 0.0275558
discuss 0.0145918	**case** 0.0133859
document 0.0112008	market 0.00471767
exchang 0.00453029	*export* 0.00372615
stock 0.00343644	close 0.00283588
talk 0.00253939	*import* 0.00227241
busi 0.00222684	report 0.00213634
deal 0.00210441	*draft* 0.00207169
negoti 0.0020529	*sell* 0.00204515
law 0.00203561	time 0.00196475

The terms in italic are the terms that are most different between the two tables. We see that the unrelated words such as time, thing, part, work, etc. are now attributed lower probabilities and even removed from the expansion terms, while several other related terms, such as law, import, export, negotiation are added.

Globally, our experiments show that when inference is applied, either to document expansion or to query expansion, the retrieval effectiveness is generally increased. This result is confirmed on several test collections in two different languages. These experiments tend to validate our initial claim that

—Inference can improve IR, and

—LM is a reasonable framework to implement inference efficiently.

7. RELATED WORK

The integration of term relationships in LM was the subject of Berger and Lafferty [1999]. Our basic approach on document expansion is similar to theirs, but we use more term relationships. In this article, we further extended the approach to query expansion. In addition, we also integrated into the models several other aspects related to inference such as specificity and multistep inference. Therefore, our approach is a substantial extension of Berger and Lafferty [1999].

In an attempt to integrate term relationships into a probabilistic framework, Turtle and Croft [1990] proposed a model based on a Bayesian network. The basic idea is comparable to that of our models: one aims to infer the relationship between a document and a query through relationships between terms. Even though the frameworks used in our work and in that of Turtle and Croft are different (LM vs. Bayesian network), it is possible to make a close comparison between them. Indeed, the Bayesian network used in Turtle and Croft [1990] contains several layers, including a term layer for terms in documents and a concept layer for terms contained in a query. The inference operation is enabled by the possible connections between document terms and query terms. This is similar to our term relationship $P_R(w|w')$, but, in Bayesian networks, it is not restricted to one single term in the condition part and can be $P(w|w_1, w_2, \dots)$. However, the complexity of the model increases exponentially with the number of terms in the condition part. In their implementation, Turtle and Croft also limited the dependencies to simple ones for efficiency. The conditional

probability $P(w|w_1, w_2, \dots)$ also consider all the combinations of presence and absence of terms w_1, $w_2, \dots$ In our models, we only consider their presence, making the models simpler.

Pseudorelevance feedback has been used in many IR experiments, for example, Zhai and Lafferty [2001b] and Kurland et al. [2005]. In most of them, feedback documents are used to determine another term distribution, which is then used in combination with the original query. In our approach, we also extracted term relationships from feedback documents, creating a set of local term relationships that are more related to the query. Our experiments show that this extension can further improve retrieval effectiveness.

Markov chain has been used in several previous studies of LM in IR. For example, Lafferty and Zhai [2001] created a Markov chain which alternates between terms and documents, that is, an inference of term from another term has to go though documents. In our model for multistep inference, a Markov chain is created directly between terms based on their co-occurrences. Although the formulation is not the same, the basic idea is comparable. Indeed, if one can derive a strong relationship between two terms, t_1 and t_2, via some documents D in the Markov chain used in Lafferty and Zhai [2001], this is also because both $P(D|t_1)$ and $P(t_2|D)$ are strong. In this case, both t_1 and t_2 would likely occur relatively frequently in D, and we can also extract a strong co-occurrence relationship between them from D.

Various logical models have been proposed in previous studies [Crestani and van Rijsbergen 1995; van Rijsbergen 1986; Huibers et al. 1996; Nie et al. 1998; Lau et al. 2004; Lasado and Bareiro 2001]. Although these models all have great theoretical inferential power, it is difficult to implement them in an efficient way due to their complexity. In this study, the models we proposed are not comparable to these logical models in terms of inferential power. Nevertheless, our models can carry out the basic inference operation. The inferential language models have made several simplification assumptions to make inference tractable in practice. In particular, we have limited ourselves to certain types of term relationships. Different inference paths have also been considered to be independent; this allows us to sum up the probabilities inferred by all the paths.

Our implementation and experiments showed that indeed, these models with some inferential capabilities are both efficient and effective. In addition, in many logical models, the calculation of uncertainty involved many heuristics. In our case, the whole development of these models follows the theoretical framework of LM without introducing such heuristic calculation. The models we proposed represent a good compromise between theoretical inferential power and practical efficiency.

8. CONCLUSIONS

In this article, we proposed to integrate the basic inference operations of IR within the language modeling framework. Inference based on term relationships has been implemented either as document expansion or query expansion. Different from the traditional approaches using LM, smoothing is now given an inference role, that is, to infer alternative representations of a document or

query using relationships between terms. The resulting models are augmented by an inference capability.

Beside the basic inferential models for document and query expansion, we also considered several important aspects in their implementation: the consideration of specificity, the utilization of multistep inference, as well as the exploitation of pseudorelevance feedback.

Our experiments have been carried out on four TREC collections. The results showed that these inferential models are also effective in practice: The addition of inference in our models brings significant improvements to several test collections. In addition, we also showed that the improvements can be obtained in both English and Chinese collections, regardless of language. Indeed, beside the linguistic resource WordNet, term relationships have been extracted from documents according to their co-occurrences. This process can be carried out on any language.

This study demonstrates that the LM framework is suitable for implementing some inference operations in IR, due to its flexibility and its robustness to noise. The success of the proposed models also suggests that our approach to increase inferential power of IR is reasonable; we start with a robust model and try to integrate as much inference capability as possible.

This study is a first attempt to integrate inferential capability into LM. The integration proposed in this article is far from reaching its limit. We only considered simple term relationships, that is, a relationship between one term and another term. This gives rise to the problem of ambiguity, for instance, when we expand a query about java, it will virtually be expanded in all possible meanings—programming language, island, or coffee. A better way to make an inference is to consider context. If we are able to identify the meaning of the word, then the expansion will result in better expansion words. Unfortunately, word disambiguation is still a difficult operation. A less ambitious method is to consider the inference of a term from a combination of some terms, for example, Java and computer. The terms in the combination could serve mutually as context when we determine the related terms. Bai et al. [2005] have proposed an approach to query expansion using context-sensitive term relationships extracted according to information flow Song and Bruza [1998]. They showed that this type of term relationship can suggest better query expansion terms than the traditional term relationship between two single words. The integration of this new type of term relationship in the inference models is a promising avenue to explore to further improve the latter. Inference using context-sensitive term relationships could produce conclusions of higher quality.

The proposed models do not consider the user or the user's domain of expertise. These variables have a great impact on the relevance of the documents retrieved. There are several interesting extensions of LM in this direction. For example, Liu and Croft [2004] proposed a LM using document clustering: a document is considered to be in its domain formed by the cluster. Document cluster can be used to improve the document expansion model as follows. We can extract term relationships from each document cluster and apply them on the documents in the cluster. By doing this, we could apply more relevant term relationships to expand documents. The same idea can be applied on a query

model. One can constitute a set of documents in the user's domain of interest of (e.g., by gathering all the documents that the user has read for one type of task or in one area) and use them to construct a domain model and to extract domain-dependent term relationships. It is also possible to use a Web directory (e.g., ODP[1]) to identify a set of documents in each of the identified categories and use them to construct a domain LM or to extract domain-related term relationships. We are also testing this approach, and our preliminary results show very encouraging results.

REFERENCES

BAI, J., SONG, D., BRUZA, P., NIE, J.-Y., AND CAO, G. 2005. Query expansion using term relationships in language models for information retrieval. *ACM CIKM Conference*. 688–695.

BERGER, A. AND LAFFERTY, J. 1999. Information retrieval as statistical translation. *ACM SIGIR Conference*. 222–229.

BRÉMAUD, P. 1999. *Markov Chains: Gibbs Fields, Monte Carlo Simulations, and Queues*. Springer-Verlag.

BROWN, P. P. F., PIETRA, S. A. D., PIETRA, V. D. J., AND MERCER, R. L. 1993. The mathematics of machine translation: Parameter estimation. *Computat. Linguist. 19*, 263–312.

BRUZA, P. D., SONG, D., AND WONG, K. F. 2000. Aboutness from a commonsense perspective. *J. Amer. Soc. Inform. Sci. Techn. 51*, 12, 1090–1105.

BRUZA, P. AND HUIBERS, T. W. C. 1996. A study of aboutness in information retrieval. *AI Rev. 10*, 5–6, 381–407.

CAO, G. NIE, J. Y., AND BAI, J. 2005. Integrating word relationships into language models. *ACM SIGIR Conference*. 298–305.

CHEN, S. F. AND GOODMAN, J. 1998. An empirical study of smoothing techniques for language modeling. Tech. rep. TR-10-98, Harvard University.

CRESTANI, F. AND VAN RIJSBERGEN, C. J. 1995. Information retrieval by logical imaging. *Document. J. 51*, 3–17.

CROFT, W. B., LUCIA, T. J., AND COHEN, P. R. 1988. Retrieving documents by plausible inference: A preliminary study. *ACM SIGIR Conference*. 481–494.

DEERWESTER, S. C., DUMAIS, S. T., LANDAUER, T. K., FURNAS, G. W., AND HARSHMAN, A. 1990. Indexing by latent semantic analysis. *J. Amer. Soc. Inform. Sci. 41*, 6, 391–407.

DEMPSTER, A. P, LAIRD, N. M., AND RUBIN, D. B. 1997. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statist. Soc. 39*, 1–38.

FOO, S. AND LI, H. 2004. Chinese word segmentation and its effect on information retrieval. *Inform. Process. Manag. 41*, 1, 161–190.

HUIBERS, T. W. C., LALMAS, M., AND VAN RIJSBERGEN, C. J. 1996. Information retrieval and situation theory. *SIGIR Forum 30*, 1, 11–25

KURLAND, O., LEE, L., AND DOMSHLAK, C. 2005. Better than the real thing? Iterative pseudo-query processing using cluster-based language models. *ACM SIGIR*. 19–26.

LAFFERTY, J. AND ZHAI, C. 2001. Document language models, query models, and risk minimization for information retrieval. *ACM SIGIR Conference*. 111–119.

LALMASM, M. AND BRUZA, P. D. 1998. The use of logic in information retrieval modeling. *Knowl. Eng. Rev. 13*, 3, 263–295.

LASADO, D. E. AND BARREIRO, A. 2001. A logical model for information retrieval based on propositional logic and belief revision. *Comput. J. 44*, 5, 410–424.

LAU, R. Y. K., BRUZA, P., AND SONG, D. 2004. Belief revision for adaptive information retrieval. *SIGIR 2004*. 130–137.

LAVRENKO, V. AND CROFT, W. B. 2001. Relevance-based language models. *SIGIR 2001*.

LEWIS, D. 1973. *Counterfactuals*. Harvard University Press.

LIU, X. AND CROFT, W. B. 2004. Cluster-based retrieval using language models. *ACM SIGIR Conference*. 186–193.

[1]Open Directory Project, http://dmoz.org/.

MANDALA, R., TOKUNAGA, T., AND TANAKA, H. 1998. Ad hoc retrieval experiments using WordNet and automatically constructed theasuri. In *Proceedings of the 7th Text Retreival Conference (TREC-10)*. 475–481.

MILLER, G. ED. 1990. Wordnet: An online lexical database. *Int. J. Lexicography 3*, 4.

MILLER, D., LEEK, T., AND SCHWARTZ, R. M. 1999. A hidden Markov model information retrieval system. *ACM SIGIR Conference*. 214–222.

NIE, J. Y. 1990. A general information retrieval model based on modal logic. *Inform. Process. Manag. 25*, 5, 477–491.

NIE, J. Y. AND LEPAGE, F. 1998. Toward a broader model for information retrieval. In *Information Retrieval, Uncertainty and Logics*. M. Lalmas, F. Crestani, C. J. van Rijsbergen, Eds. Kluwer Academic Publishers. 17–38.

NIE, J. Y. 2003. Query expansion and query translation as logical inference. *J. Amer. Soc. Inform. Sci. Techn. 54*, 4, 335–346.

PONTE, J. AND CROFT, W. B. 1998. A language modeling approach to information retrieval. *ACM SIGIR Conference*. 275–281.

SONG, D. AND BRUZA, P. D. 1998. Towards context-sensitive information inference. *J. Amer. Soc. Inform. Sci. Techn. 54*, 4, 321–334.

TOUTANOVA, K., MANNING, C. D., AND NG, A. Y. 2004. Learning random walk models for inducing word dependency distributions. *ICML Conference*. 103–110.

TURTLE, H. AND CROFT, W. B. 1989. Inference networks for document retrieval. *ACM SIGIR Conference*. 1–24.

VAN RIJSBERGEN, C. J. 1986. A non-classical logic for information retrieval. *Comput. J. 29*, 6, 481–485.

VOORHEES, E. 1994. Query expansion using lexical-semantic relations. *ACM SIGIR Conference*. 61–69.

WONG, K. F., SONG, D., BRUZA, P. D., AND CHENG, C. H. 2001. Application of aboutness to functional benchmarking in information retrieval. *ACM Trans. Inform. Syst. 19*, 4, 337–370.

XU, J. AND CROFT, W. B. 1996. Query expansion using local and global document analysis. *ACM SIGIR Conference*. 4–11.

ZHAI, C. AND LAFFERTY, J. 2001a. A study of smoothing methods for language models applied to information retrieval. *ACM SIGIR Conference*. 334–342.

ZHAI, C. AND LAFFERTY, J. 2001b. Model-based feedback in the language modeling approach to information retrieval. *ACM CIKM Conference*. 403–410.

ZHAI, C. AND LAFFERTY, J. 2002. Two-stage language models for information retrieval. *ACM SIGIR Conference*. 49–56.

Received December 2005; revised June 2006; accepted October 2006