Word Pairs in Language Modeling for Information Retrieval

Carmen Alvarez, Philippe Langlais and Jian-Yun Nie
Dept. IRO, Université de Montréal
CP. 6128, succursale Centre-ville
Montréal, Québec, H3C CJ7 Canada
{bissettc,felipe,nie}@iro.umontreal.ca

Abstract

Previous language modeling approaches to information retrieval have focused primarily on single terms. The use of bigram models has been studied, but the restriction on word order and adjacency may not be justified for information retrieval. We propose a new language modeling approach to information retrieval that incorporates lexical affinities, or pairs of words that occur near each other, without a constraint on word order. The use of compound terms in the vector space model has been shown to outperform the vector model with only single terms (Nie & Dufort, 2002). We explore the use of compound terms in a language modeling approach, and compare our results with the vector space model, and unigram and bigram language model approaches.

1 Introduction

Traditional approaches to information retrieval include the boolean model, the vector space model, and probabilistic models of relevance (Fuhr, 1992), (Hiemstra & de Vries, 2000). These approaches include a model for indexing, or an abstraction on how documents and queries are represented, as well as a relevance model, which defines a scoring function between the document and query representations. Collection statistics such as word frequency, document frequency and document length, are included in the indexing and scoring in a heuristic manner.

Language modeling is another family of information retrieval approaches that do not formulate relevance directly. Instead, a language model M_d is trained for a document d, and the score of the document is determined by the probability of generating the query q given this document model, or $p(q|M_d) = p_d(q)$. Previous experiments have shown that language models can perform at least as well as, and in many cases better than, the classical models. In addition, language modeling offers a framework to information retrieval, capable of generating other traditional models.

However, the majority of previous language modeling approaches have typically assumed that query terms are independent (i.e., $p_d(\text{language model}) = p_d(\text{language}) \times p_d(\text{model})$). This assumption is too strong, and in this paper, we try to consider dependencies between query

terms. Rather than use classical bigrams, in which adjacency and word order are over-stressed for information retrieval purposes, we propose approaches that use word pairs but relax these constraints. Our experiments show that word pairs can improve the effectiveness of a unigram model.

2 Previous work

Ponte and Croft (1998) proposed the first language modeling approach to information retrieval. Each document is considered to be a sample of a particular language, and a language model is trained for each document. For a given query, the documents are sorted in decreasing order of the probability that the document language model generates the query. This language modeling approach combines the indexing and the scoring function in a single model. In addition, the collection statistics are implicit in the model. In Ponte and Croft's initial approach, no heuristic parameters are learned or fine-tuned to improve performance; the model is based strictly on the observed data, including the relative frequencies of words in a document, the global frequencies of the words in the collection, and the average frequencies of words among the documents in which they occur. An important process in this model is the smoothing of the document language models, as the sparse data problem is significant for documents, which are typically quite short.

Song and Croft (1999) improve this initial approach with a new language model. Several smoothing techniques are presented, including a Good-Turing estimate of the word counts in a document, and interpolated combinations with a corpus and bigram document models. The relative contributions of the different models (document and corpus unigram models, and bigram models) to the query generation probability are determined empirically.

Hiemstra (2002) also proposes an interpolation between the unigram document language models and the corpus model. Rather than determine the relative contributions of the models empirically, the weighting coefficients are hidden variables denoting the importance of the terms in the query. An instance of the EM algorithm is used to automatically learn the importance of query terms, and relevance feedback is used to provide the set of relevant documents used by the algorithm.

Lavrenko and Croft (2001) incorporate language models in the classic probabilistic approach. The primary challenge to the classic probabilistic approach is estimating a relevance model, or the probability that a word belongs to the relevant class, with no training data indicating the relevant documents. Lavrenko and Croft use document and corpus language models to estimate the relevance model, without the need for relevance training data.

Zhai and Lafferty (2001) introduce the idea of a query language model and sort the documents in decreasing order of the similarity between the document model and the query model. Jin

et. al. (2002) and Berger and Lafferty (1999),(Lafferty & Berger, 1999) propose a translation approach to information retrieval. The documents are considered to be samples of a verbose language, and the queries samples of a concise language. A document-query translation model is constructed from the collection and the documents are sorted according to the probability that the query is a translation of the document.

Several approaches have attempted to account for word dependancies. Jiang *et. al.* (2004) incorporate bigram phrases in a backoff combination with unigrams, but the results do not outperform Song and Croft's (1999) linear interpolation between conditional bigrams and unigrams. Srikanth and Srihari (2002) develop a biterm model, modeling probabilities of adjacent word pairs without consideration of the order of the words, and report slight improvements over Song and Croft's conditional bigram approach. Pickens (2000) introduces an approach that uses non-adjacent biterms, but the particular domain (musical documents) requires an emphasis on the order of "words" (musical notes represented as text).

In these existing approaches, either an assumption of independence between query words is made, or restrictions on word order and/or adjacency of word pairs exists. While the independance assumption simplifies the relevance score calculation, it is an oversimplification of the problem of information retrieval. Query words often have dependencies, such as in the query "language model". We believe that better performance may be achieved in modeling these word dependencies, without word order or adjacency constraints, as the purpose of information retrieval is not to find the same word sequences of the query in the documents. However, we do expect that words in a relevant document will have similar dependencies as in the query. For the example "language model", it is not necessary to find documents with the same bigram, but rather to find documents in which "language" is related to "model" in some way, as a pair. Some possible examples are "modeling languages" or "modeling a document language".

We propose a language modeling approach that incorporates word pairs, without a constraint on adjacency or word order. One important question is which pairs to consider. In our approach, we select word pairs by determining statistical relationships, or lexical affinities, between words. Our experiments show that these word pairs improve information retrieval effectiveness.

Before exploring the use of word pairs, we present several smoothing techniques essential for unigram and bigram models in information retrieval, in sections 3 and 4. We then describe in section 5 two models that use lexical affinities: first, a model (AL-1) based primarily on pairs, which are decomposed into their constituent words for smoothing, followed by a second model (AL-2) that includes the single words of a document as well as its word pairs which are not broken down for smoothing. Our experiments and a discussion about the results for each model are given in sections 6 and 7, followed by a conclusion and thoughts on future work in section 8.

3 Relevance based on a unigram model

A relevance score for a document d and a query q, based on the probability that an n-gram model for the document generates the query $q = w_1, \ldots, w_N$ is as follows:

$$score(d, q) = p_d(q) = p_d(w_1) \prod_{i=2}^{N} p_d \left(w_i | w_{max(1,i-n+1)}^{i-1} \right) \tag{1}$$

In the unigram case, the score is simply the product of the individual query term probabilities according to the document model:

$$score(d, q) = \prod_{i=1}^{N} p_d(w_i) \tag{2}$$

while for a bigram, the score includes a context of one word:

$$score(d, q) \quad = \quad p_d(w_1) \prod_{i=2}^{N} p_d(w_i | w_{i-1}) \tag{3}$$

The probability that a single word w is generated by the language model for document d, according to the maximum likelihood estimate, is its relative frequency:

$$p_{MLE}(w) = \frac{c(w)}{N} \tag{4}$$

where $c(w)$ is the number of times the word w appears in the document, and N is the total number of words in the document. Using the maximum likelihood estimate directly in equation 2 for $p_d(w)$ would result in relevance scores of zero for documents which do not contain one or more query terms. Since documents typically provide a rather small sample of data from which to train a model (a typical document containing about 200 - 400 words), this sparse data problem is more pronounced for information retrieval than for other language modeling applications where a single corpus of millions of words is used for training. Smoothing becomes essential.

The simplest smoothing approach to account for unobserved words is to add a token to the vocabulary for unknown words, unk, and assign a fixed probability, $p_d(unk)$ to all words not present in the document d. Intuitively, $p(unk)$ should be lower than the observed words of a given document, to prevent an unseen word in one document from having a greater probability than an observed word in another, but this parameter may be determined empirically. The probabilities of the observed words must then be discounted:

$$p_d(w) = p_{d_{fixed}}(w) = p_{d_{MLE}}(w) \times (1 - p_d(unk)) \tag{5}$$

The fixed smoothing approach resolves the problem of a zero-probability for documents. However, not all unseen words should receive the same probability. Song and Croft (1999) give the example of a document collection specialized in the domain of information retrieval, and a query containing the words "keyword" and "crocodile". In such a collection, the word "keyword" would occur very frequently, while "crocodile" is likely to have a low document frequency. Intuitively, if "keyword" is not present in a document, its contribution to the relevance score should not be equal to the contribution of a missing word "crocodile".

One smoothing technique that addresses the relative contributions of unknown words consists of interpolating the probabilities of the document model with a corpus model as follows:

$$p_d(w) = p_{d_{corpus\_interp}}(w) = \lambda_d p_{d_{MLE}}(w) + (1 - \lambda_d) p_{corpus_{MLE}}(w)) \tag{6}$$

The optimal contribution of words observed in the document d, λ_d, must be determined empirically, or through some training process such as an Estimation Maximization (EM) algorithm.

It is possible that a query contains a word that never occurs in the collection. To prevent all documents from receiving a zero score, we smooth the corpus model with a fixed probability $p_{corpus}(unk)$ similar to the initial technique used for document models:

$$p_{corpus\_fixed}(w) = p_{corpus_{MLE}}(w) \times (1 - p_{corpus}(unk)) \tag{7}$$

and the modeled document probability becomes:

$$p_d(w) = p_{d_{corpus\_fixed}}(w) = \lambda_d p_{d_{MLE}}(w) + (1 - \lambda) p_{corpus\_fixed}(w) \tag{8}$$

4 Relevance based on a bigram model

The assumption of independence between words, used in the unigram model as well as a large part of information retrieval approaches, is not always justified. For example, if a user specifies the request "information retrieval", a document that discusses search engines and contains the term "information retrieval" is intuitively more relevant than a document unrelated to the domain with the words "information" and "retrieval" in independent contexts.

One possible way to consider the dependency between words is to estimate the probability that one word follows another, with a bigram model $p_d(w_i|w_{i-1})$, using equation 3 for the relevance

score. The maximum likelihood estimate of a bigram probability is its relative frequency, which is the frequency of the bigram divided by the number of times the history, w_{i-1} occurs:

$$p_{d_{MLE}}(w_i|w_{i-1}) = \frac{c_d(w_{i-1}w_i)}{\sum_w c_d(w_{i-1}w)} = \frac{c_d(w_{i-1}w_i)}{c(w_{i-1})} \quad (9)$$

Our modeled bigram probability includes the maximum likelihood estimate smoothed with a unigram model in the following interpolation:

$$p_d(w_i|w_{i-1}) = \lambda_2 p_{d_{MLE}}(w_i|w_{i-1}) + (1-\lambda_2)p_d(w_i) \quad (10)$$

where the unigram model $p_d(w)$ uses either the fixed smoothing (eq. 5) or an interpolation with the corpus unigram model (eq. 8). In this approach, one would naturally raise the following question: should all the bigrams be considered (with constraints on word order and adjacency) in information retrieval, given the high cost of their estimation? In the following section we present a new approach to incorporating word pairs in language models for IR, without these constraints, and we compare the results with the bigram approach in section 6.

5 Lexical Affinities

5.1 Introduction

Bigram and trigram models attempt to account for the dependency between terms by taking the context of words into consideration. However, these models assume that the order of words is important. While valid for applications such as speech recognition, this assumption does not necessarily apply to information retrieval. The query "apartment rentals" is one particular example. After the preprocessing stage, the request becomes "apartment rent". A document containing the phrase "rent an apartment" should not be considered a priori any less relevant than a document with the phrase "apartments for rent."

Another lexical unit which takes context into consideration without a restriction on word order is the lexical affinity, a pair of words that co-occur in a given text, separated by at most n words. Martin *et. al.* (1983) suggest that 98% of lexical relations are between words within a window of 5 plain words, and this is the distance that we use.

Maarek *et. al.* (1991) introduce the concept of the resolving power of a pair of words. Like the *idf* factor used in the vector model, or the integration of a corpus model with a unigram model, the principal idea behind the resolving power is that the importance of a word for a given document should be based not only on the frequency of the word in the document, but also

its frequency in the entire collection. One of the goals is to prevent non-discriminating words (which may be stoplist words such as "a" or "the", or other words which have a high document frequency) from having too much relative weight in the document's representation. The lexical affinities which best characterize a document are those which have both a high frequency in the document and a relatively low frequency in the collection. The information, or discrimination power, of a word pair $< u, v >$ is inversely proportional to the global frequencies of its words u and v. The authors make a hypothesis of independence between words in the global context and introduce the following representation of the information of a pair $< u, v >$:

$$\text{INFO}(< u, v >) = -\log(p_{corpus}(u) \times p_{corpus}(v)) \tag{11}$$

The resolving power of the pair, $\rho_d(< u, v >)$, for a document d, includes the frequency of the pair in the document as well as its information:

$$\rho_d(< u, v >) = c_d(< u, v >) \times \text{INFO}(< u, v >) \tag{12}$$

where $c_d(< u, v >)$ is the frequency of the pair $< u, v >$in the document d.

5.2 Model AL-1: a language model based on lexical affinities

We describe here a language model that incorporates lexical affinities.

5.2.1 Estimating the probabilities of word pairs

A language model based on lexical affinities may be constructed and used for retrieval in a similar way as the unigram model described in section 3. If a word pair $< u, v >$ is considered as a single lexical unit, the probability of seeing the pair in a document d is its relative frequency, calculated in the same way as for the unigram model (eq. 4):

$$p_{d_{MLE,freq}}(< u, v >) = \frac{c_d(< u, v >)}{\sum_{u,v} c_d(< u, v >)} \tag{13}$$

The discrimination power of the pair may be incorporated in the model by using the resolving power as a modified frequency:

$$p_{d_{MLE}}(< u, v >) = \frac{\rho_d(< u, v >)}{\sum_{u,v} \rho_d(< u, v >)} \tag{14}$$

The sparse data problem has a greater impact on a probabilistic model based on lexical affinities than on a unigram model. There are two types of unobserved events: a pair $< u, v >$, where u and v have been observed in the document d, but never within the specified window, and the pair $< unk, w >$, where one or possibly two of the words are not present in the document. Intuitively, the first pair should have a greater probability than the second. The probabilities of the single words allow us to capture this information. We smooth the pair probabilities with a unigram model for the individual words as follows:

$$p_d(< u, v >) = \lambda_2 p_{d_{MLE}}(< u, v >) + \theta_{u,v} \left[\alpha \frac{p_d(u)}{|V|} + \beta \frac{p_d(v)}{|V|} \right] \tag{15}$$

$$\theta_{u,v} = \begin{cases} 1 & \text{if } p_{d_{MLE}}(< u, v >) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where $|V|$ is the number of words in the vocabulary and $p_d(u)$ and $p_d(v)$ are calculated according to the unigram model as described in section 3. Note that $\sum_{u,v} \frac{p_d(u)}{|V|} = 1$. Since there is no reason a priori to assign more weight to the word u than to v, we can let $\alpha = \beta$ and $\gamma = \frac{\alpha}{|V|} = \frac{\beta}{|V|}$, and simplify the equation as follows:

$$p_d(< u, v >) = \lambda_2 p_{d_{MLE}}(< u, v >) + \theta_{u,v} \gamma \left(p_d(u) + p_d(v) \right) \tag{17}$$

γ is then solved in terms of λ_2 so that the probabilities of all pairs sum to 1:

$$\begin{aligned} \sum_{u,v} p(< u, v >) = 1 &= \sum_{u,v} \lambda_2 p_{d_{MLE}}(< u, v >) + \sum_{u,v} \theta_{u,v} \gamma \left(p_d(u) + p_d(v) \right) \\ 1 &= \lambda_2 + \gamma \sum_{u,v} \theta_{u,v} \left(p_d(u) + p_d(v) \right) \end{aligned}$$

$$\gamma = \frac{1 - \lambda_2}{\sum_{u,v} \theta_{u,v} \left(p_d(u) + p_d(v) \right)}$$

We will describe two approaches to incorporate lexical affinities in the calculation of relevance of a document d for a query q. First, we consider a relevance score based on the probability of generating the query, according to the document model, in a similar fashion to the unigram model discussed in section 3. Then we propose a score based on the Kullback-Leibler divergence, which measures the distance between two distributions. In our case, the distance is measured between a document model and a query model. We show that the KL-divergence is similar to the probability of generating the query, but incorporates more information from the query.

5.2.2 Probability of generating the query

Since our model AL-1 treats a pair as a complete lexical unit, it is comparable to a unigram model, and the probability of generating the query q from a model for document d is the product of the probabilities of each pair in the query:

$$score_{gen}(d,q) = p_d(q) = \prod_{<u,v>\in q} p_d(<u,v>)^{c_q(<u,v>)} \tag{18}$$

where $p_d(<u,v>)$ is the probability of the pair $<u,v>$ as determined by equation 15, and $c_q(<u,v>)$ is the number of times the pair is observed in the query q. As with a unigram model, we assume a hypothesis of independence between the lexical units (independence between pairs in this case).

5.2.3 KL-divergence

The Kullback-Leibler divergence $D(P||Q)$, also known as relative entropy, is a measure of distance between two probabilistic distributions $P(x)$ and $Q(x)$ over the same set of events. Relative entropy is also a measure of the information loss if the distribution Q is used as an approximation of an observed distribution P. The KL-divergence between the two distributions is as follows:

$$D(P||Q) = \sum_x P(x) \times \log\left(\frac{P(x)}{Q(x)}\right) \tag{19}$$

In our context, the observed distribution is the query model and the goal is to sort the documents in decreasing order of the divergence between the query model and that of the document. Therefore,

$$D(q||d) = \sum_{a\in q} p_q(<u,v>) log\left(\frac{p_q(<u,v>)}{p_d(<u,v>)}\right) \tag{20}$$

The relevance score of a document d for a query q is then:

$$\begin{aligned} score_{KL}(d,q) &= -|D(q\|d)| \\ &= -\left|\sum_{<u,v>\in q} p_q(<u,v>)\left(\log p_q(<u,v>) - \log p_d(<u,v>)\right)\right| \end{aligned} \tag{21}$$

The closer a document model is to the query model, the closer the score approaches zero.

We can show that the KL-divergence is similar to the probability of generating the query, but incorporates more information from the query. Given the relative sizes of a document and a query, in most, if not all, cases, the probability of any given pair will be higher in the query model than in the document model. The divergence is then always positive:

$$\begin{aligned} score_{KL}(d,q) &= -\left[\sum_{<u,v>\in q} p_q(<u,v>)\left(\log p_q(<u,v>) - \log p_d(<u,v>)\right)\right] \\ &= -\sum_{<u,v>\in q} p_q(<u,v>)\log p_q(<u,v>) \\ &\quad + \sum_{<u,v>\in q} p_q(<u,v>)\log p_d(<u,v>) \end{aligned} \tag{22}$$

Since the first term is identical for all documents, we may remove it from the relevance score:

$$\begin{aligned} score_{KL}(d,q) &= \sum_{<u,v>\in q} p_q(<u,v>)\log p_d(<u,v>) \\ &= \log\left(\prod_{<u,v>\in q} p_d(<u,v>)^{p_q(<u,v>)}\right) \end{aligned} \tag{23}$$

The only difference between this formula and the probability of generating the query is that here, the pair is included in the product $p_q(<u,v>)$ times rather than $c_q(<u,v>)$ times. Thus, the resolving powers of the pairs in the query are included in the score, rather than their frequencies.

5.3 Model AL-2: lexical affinities as atomic lexical units

In contrast to our model AL-1 which is based primarily on lexical affinities whose individual terms are used only for smoothing, we may construct a simple unigram model which contains the single words in the document as well as the pairs represented as single units. For example, the pair containing the words "famine" and "sudan" is added to the unigram model as the word "famine,sudan". The individual words "famine" and "sudan" are also included in the distribution. The observed counts of words and pairs are modified and we search the optimal relative contributions of each. Specifically, the modified counts of a word w and a pair $<u,v>$ in a document d are as follows:

$$\begin{aligned} c_d^*(w) &= \alpha_d c_d(w) \\ c_d^*(<u,v>) &= \beta_d c_d(<u,v>) \end{aligned} \tag{24}$$

Rather than use the observed counts of the pairs, we may base the modified counts on the resolving power as follows:

$$c_d^*(< u, v >) = \beta_d \rho_d(< u, v >) \tag{25}$$

The word and pair probabilities are then determined according to the relative frequencies:

$$p_{d_{MLE}}(w) = \frac{c_d^*(w)}{\sum_{w'} c_d^*(w')}$$
$$p_{d_{MLE}}(< u, v >) = \frac{c_d^*(< u, v >)}{\sum_{w'} c_d^*(w')} \tag{26}$$

where w' is either a single word w or a pair $< u, v >$ represented as a word. The same smoothing techniques described in section 3 may be applied to this model AL-2. For example, a fixed probability for unobserved terms $p_d(unk)$ may be applied as in equation 5, or a corpus model $p_{corpus}(w')$ may be integrated as in equation 8. Specifically, for the fixed smoothing, we have:

$$p_{d_{fixed}}(w') = p_{d_{MLE}}(w') \times (1 - p(unk)) \tag{27}$$

and the interpolation with the corpus model is:

$$p_{d_{corpus\_interp}}(w') = \lambda_d p_{d_{MLE}}(w') + (1 - \lambda_d) p_{corpus\_fixed}(w')) \tag{28}$$

The relevance score of a document d for a query q is the probability of generating the query according to the document model:

$$score(d, q) = p_d(q) \quad = \quad \prod_{w' \in q} p_d(w')^{c_q(w')} \tag{29}$$

We may also optimize the relative contributions of the words and pairs in the query, and modify their frequency with factors α_q and β_q as with the documents. The score then becomes:

$$score(d, q) = \prod_{w' \in q} p_d(w')^{c_q^*(w')} \tag{30}$$

6 Experimental results

We now present the results we have observed for the various techniques discussed. The experiments are performed on the AP90 TREC collection, which contains 78321 documents in English from Associated Press newswire articles from 1990. For each experiment, documents are retrieved for 53 queries from the TREC-6 and TREC-7 cross language topics in English. Each experiment includes a test using just the title field of the requests, which contain 1-5 words (2.5 words on average), as well as a test using the title and description fields (3-19 words, with an average of 7 words). Both the documents and the queries undergo a preprocessing stage which consists of stemming and removing stop words with a stoplist. The SMART system is used for the vector space model, with the ltc weighting scheme.

run	equation for $p_d(w)$	titles	titles and descriptions
vector space model		0.3349	0.3498
$p_{d_{fixe}}(w) = 10E-3$	5	0.3536	0.3464
$p_{d_{fixe}}(w) = 10E-4$	5	0.3574	0.3440
$p_{d_{fixe}}(w) = 10E-5$	5	0.3639	0.3226
$p_{d_{fixe}}(w) = 10E-6$	5	0.3637	0.2980
$p_{d_{fixe}}(w) = 10E-7$	5	0.3637	0.2982
$p_{d_{fixe}}(w) = 10E-2$	5	0.1633	0.1210
$p_{corpus_{MLE}}(w), \lambda_d = 0.1$	6	0.3799	0.3592
$p_{corpus_{MLE}}(w), \lambda_d = 0.2$	6	0.4064	0.3783
$p_{corpus_{MLE}}(w), \lambda_d = 0.4$	6	0.4050	0.3764
$p_{corpus_{MLE}}(w), \lambda_d = 0.6$	6	0.4068	0.3663
$p_{corpus_{MLE}}(w), \lambda_d = 0.8$	6	0.3949	0.3461
$p_{corpus\_fixed}(w), \lambda_d = 0.2$	8	0.4067	0.4275
$p_{corpus\_fixed}(w), \lambda_d = 0.4$	8	0.4053	0.4241
$p_{corpus\_fixed}(w), \lambda_d = 0.6$	8	0.4071	0.4139

Table 1: Results for a relevance score based on a unigram model (equation 2), using fixed and corpus smoothing

6.1 Experiments with the unigram model

Table 1 shows the average precision obtained by the unigram model, with fixed smoothing, interpolation with a non-smoothed corpus model, and interpolation with a corpus model smoothed with a fixed $p_{corpus}(unk)$. As we can see in table 1, the unigram with fixed smoothing outperforms the vector space model for the title queries, but not for the title and description queries. Since the title queries contain on average between 2 and 3 words, it is likely that all of the query terms are present in most relevant documents, or that typically at most one query term is absent from a relevant document. For the longer queries, however, it is likely that relevant documents often do not contain two or more query terms. In this case, the same probability is assigned to all missing terms, not taking into account the relative importance of the terms.

The interpolation with the corpus model leads to superior performance to the fixed smoothing and allows the system to outperform the vector space model for both types of query. In addition, when the corpus model is smoothed with a $p_{corpus}(unk)$ to account for query terms never seen in the collection, a significant increase in performance (from 37.9% to 42.8%) is visible with the longer queries. The longer queries have a greater possibility of containing words not seen in the collection. In addition, the longer queries perform better than the title queries with corpus smoothing (when the corpus includes the token *unk*), as the relative importance of missing words is included in the relevence score.

	fixed		corpus	
	$p(unk) = 10E-3$		$\lambda_d = 0.6$	$\lambda_d = 0.2$
λ_2	titles	titles&desc	titles	titles&desc
vector space model	0.3349	0.3498	0.3349	0.3498
unigram model	0.3536	0.3464	0.4071	0.4275
0	0.3536	0.3464	0.4071	0.4275
0.0001	0.3536	0.3464	0.4072	0.4277
0.001	0.3545	0.3473	0.4070	0.4271
0.01	0.3589	0.3544	0.4067	0.4183
0.1	0.3408	0.3390	0.3930	0.4117

Table 2: Average precision with a bigram model. The unigram model is either smoothed with a fixed $p(unk)$ (first two columns) or smoothed with the corpus model (third and fourth columns)

6.2 Experiments with the bigram model

Table 2 shows the average precision obtained by a relevance score based on a bigram model (equation 3). The modeled bigram probabilities include the observed bigram counts smoothed with a unigram model, as in equation 10. In the columns titled "fixed", the unigram uses a fixed smoothing (equation 5), and in the columns titled "corpus" the unigram model is smoothed with a smoothed corpus model (equation 8). The precision of the vector space model as well as the unigram models with the same values for $p(unk)$ and λ_d are shown.

For the unigram with a fixed smoothing, the bigram model offers a slight improvement (from 35.4% to 35.9% for the title queries and from 34.6% to 35.4% for the title and description queries). For the unigram smoothed with a corpus model, there is no significant improvement. Note that when $\lambda_2 = 0$, the results are the same as for the unigram models.

6.3 Experiments with model AL-1

Table 3 shows the effect of the relative contribution λ_2 of observed pairs in the model AL-1, equation 15. In the extreme case where only the observed pairs contribute to the relevance score ($\lambda_2 = 1$), we have a sparse data problem and the precision obtained is only 10.7% for the title queries and 7% for the title and description queries. At the other extremity, if only the single words contribute to the score ($\lambda_2 = 0$), the performance is not as good as when the pair probabilities are included. However, the results are not sensitive to values of λ_2 between 0.1 and 0.9.

Table 4 shows the effectiveness of the two relevance scores for model AL-1: the probability that the document model generates the request, and the KL-divergence. In these experiments, $\lambda_2 = 0.1$. The results of the two approaches are comparable, with query generation performing slightly better for the title queries, and KL-divergence performing slightly better than query generation

λ_2	titles	titles& descr
0.1	0.2587	0.2903
0.5	0.2575	0.2899
0.9	0.2568	0.2860
0	0.2222	0.2847
1	0.1072	0.0695

Table 3: Average precision obtained with the model AL-1, with several values for λ_2. The relevance score is the probability of generating query, equation 18.

relevance score	equation	titles	titles and descriptions
generate the query	18	0.2587	0.2903
kl-divergence	21	0.2512	0.3034
vector space model		0.3349	0.3498

Table 4: Average precision obtained by generating the query and KL-divergence for the relevance score.

for the title and description queries. However, neither of these approaches outperforms the vector space model or the unigram model.

6.4 Model AL-2: lexical affinities as atomic lexical units

6.4.1 Fixed smoothing

Table 5 shows the average precision obtained by the model AL-2. The relevance score is equation 30, and the resolving power of pairs is used for the modified counts (equation 25). In the documents, only the pairs with a resolving power $\rho > \bar{\rho} + \sigma$ (where $\bar{\rho}$ is the mean and σ is the standard deviation of the resolving powers in the document) are counted, while all pairs in the queries are used. A fixed smoothing $p(unk) = 10E-4$ is used for unknown terms (equation 27). The combinations of β_d and β_q for which this model outperforms both the vector space model and the unigram model that does not include pairs are indicated with a star (*). Results which outperform both the vector and bigram models are indicated with a dagger (†). Our model outperforms both the best unigram and best bigram models for several combinations of $\beta_d = 0.05, 0.1$ and β_q between 0.01 and 0.1.

For both types of query, the performance drops if the relative contribution of the pairs is too large. For the title queries, which contain 2.2 pairs on average (refer to table 7 for statistics on words and pairs in the queries), the performance drops noticeably for $\beta_d \geq 0.5$ and $\beta_q \geq 0.5$, with the lowest precision of 31.5% for $\beta_d = 1$ and $\beta_q = 0.5$. The title and description queries however, which contain about 8 times as many pairs as the short queries (18.2 pairs on average), are more sensitive to the contribution of the pairs in the query, β_q. The decrease in performance is generally visible for $\beta_d \geq 0.5$ and $\beta_q \geq 0.3$, and drops to 28.1% for $\beta_d = 0.3$ and $\beta_q = 1$.

	titles				titles and descriptions			
β_q / β_d	0.01	0.05	0.1	0.5	0.01	0.05	0.1	0.3
0.01	0.3585	0.3612†	0.3657*†	0.3531	0.3457	0.3497*	0.3571*†	0.3409
0.04	0.3488	0.3640*†	0.3637†	0.3494	0.3485*	0.3560*†	0.3539*	0.3163
0.1	0.3477	0.3642*†	0.3600†	0.3496	0.3506*	0.3585*†	0.3478	0.3026
0.5	0.3466	0.3462	0.3510	0.3282	0.3545*†	0.3501*	0.3339	0.2884
1	0.3340	0.3346	0.3387	0.3145	0.3517*	0.3464	0.3276	0.2810
vector space model	0.3349				0.3498			
best 1-gram w/o pairs	0.3639				0.3464			
best 2-gram	0.3589				0.3544			

Table 5: Average precision obtained by the model AL-2 (equation 30). In these experiments, the resolving power of pairs is used for the modified counts (equation 25). The relative weight of single words remains constant ($\alpha_d = 1$ and $\alpha_q = 1$) and a range of values for the relative weight of pairs (β_d,β_q) are tested. Fixed smoothing is used for all models.

For the title queries, the largest gain over the bigram model is from 35.89% to 37.01%, a relative gain of 3.1%, and the best precision obtained with the title and description queries is from 35.44% to 36.11%, a relative gain of 1.9%.

6.4.2 Smoothing with a corpus model

In table 6 we show the average precision obtained by the model AL-2, when it is smoothed with a corpus model, as described in section 5.3. The integration of the corpus model improves performance, and for the title queries, our model outperforms the best unigram and bigram models that also use a corpus model for smoothing, for several combinations of β_q and λ_d. For the title queries, the largest gain over the bigram model is from 40.72% to 41.54%, a relative gain of 2.0%. The best precision obtained with the title and description queries (43.20%) has a relative gain of 1.0% over the best bigram (42.77%) model. We can see that, like the unigram model, the longer queries perform better than the title queries, if a corpus model is used to account for the relative importance of missing query terms, and worse if a fixed smoothing is used for all unknown words. Results which are significant according to the Wilcoxon signed-rank test, with a 95% confidence interval, are indicated in bold.

6.4.3 Examples

Tables 8 and 9 illustrate how the model AL-2 may increase (table 8) or decrease (table 9) the precision of the documents returned, in comparison with a unigram model that includes only single words. To simplify the illustration, the smoothing in these examples is fixed, with $p(unk) = 0.0001$, and the weighting term of pairs in the query, β_q, is 1. Equation 27 is used to

	titles $\beta_q = 0.1$		titles and descr. $\beta_q = 0.05$	
(β_d,β_q,β_{corpus})	(0.01,0.01,0.01)	(0.01,0.01,0.0001)	(0.01,0.005,0.01)	(0.04,0.005,0.0001)
λ_d				
0.1	0.3920	0.3939	0.4118	0.4095
0.2	0.4077*	0.4092*	**0.4320***	**0.4314***
0.4	0.4087*	0.4095*	0.4266	0.4274
0.5	0.4143*	0.4154*	0.4229	0.4239
0.6	0.4113*	0.4139*	0.4196	0.4200
vector space model	0.3349		0.3498	
best 1-gram w/o pairs	0.4071		0.4275	
best 2-gram	0.4072		0.4277	

Table 6: Average precision obtained by the model AL-2 (equation 28), with several values of λ_d and β_d. The relative weight of words remains constant ($\alpha_d = 1$ and $\alpha_q = 1$). The unigram models are smoothed with a corpus model (equation 8).

	titles	titles and descr.
Avg. # words per query	2.4	6.5
Avg. # pairs per query	2.2	18.2

Table 7: Average number of words and pairs per query

calculate the score of a document for the query.

The document AP900301-222 has been judged relevant for title query #38. This document contains two of the three query terms ("debt" and "poland"), as well as one pair ("debt,poland"). AP900622-142 has been judged non-relevant. It contains all three query terms, but at lower relative frequencies than the relevant document, and it contains no pairs from the query. The unigram model which does not include pairs assigns a higher score to the non-relevant document, while the model AL-2 favors the relevant document, AP900301-222.

For query #5 on "acupuncture", document AP900308-239, which contains two of the three query terms ("acupuncture" and "study") has been judged relevant, while document AP901206-162, which also contains two query terms ("case" and "study") as well as one pair ("case,study") has been judged non-relevant. Intuitively, document AP900308-239 should be more relevant, containing the less common word "acupuncture". Although document AP901206-162 contains the same number of query terms as the AP900308-239, as well as a pair from the query, the words "case" and "study" are rather general, and this document probably does not satisfy the user's information need. In this example, the unigram model which does not include pairs correctly assigns a higher score to the first document. However, the presence of the pair "case,study" causes the model AL-2 to assign a higher score to the non-relevant document, AP901206-162.

Query #38:

```
title: conversion debt poland
```

	1-gram without pairs		Model AL-2	
	AP900301-222*	AP900622-142	AP900301-222*	AP900622-142
conversion	-	-2.28	-	-2.44
debt	-1.41	-2.28	-1.52	-2.44
poland	-1.11	-1.68	-1.22	-1.84
debt,poland	-	-	-1.54	-
conversion,debt	-	-	-	-
conversion,poland	-	-	-	-
score	-6.52	-5.24†	-16.28†	-18.72

Table 8: Scores for a relevant and non-relevant documents, for title query #38. A star (*) denotes a relevant document according to the relevance judgments, and a dagger (†) denotes the document to which the model assigns a higher score. Note that all values shown here are the log of the probabilities.

Query #5:

```
title: acupuncture
description: case study acupuncture
```

	1-gram without pairs		Model AL-2	
	AP900308-239*	AP901206-162	AP900308-239*	AP901206-162
acupuncture	-2.42	-	-2.56	-
acupuncture	-2.42	-	-2.56	-
case	-	-1.32	-	-1.50
study	-1.72	-1.45	-1.86	-1.63
acupuncture,case	-	-	-	-
case,study	-	-	-	-1.93
acupuncture,study	-	-	-	-
score	-10.56†	-10.77	-22.98	-21.06†

Table 9: Scores for a relevant and non-relevant documents, for title and description query #5. A star (*) denotes a relevant document according to the relevance judgments, and a dagger (†) denotes the document to which the model assigns a higher score. Note that all values shown here are the log of the probabilities.

7 Discussion

7.1 Unigram and bigram models

In the unigram and bigram models, as well as the model AL-2, the smoothing of unknown words is an important factor in the precision obtained. Using a fixed smoothing $p_d(unk)$ allows the performance to surpass the vector model in most cases, but a smoothing with the corpus model results in a much more significant improvement. In addition, further smoothing the corpus model with a fixed $p_{corpus}(unk)$ allows a significant improvement for the title and description queries. For these longer queries, 2% of the words, on average, are not present in the collection. The precision as well as recall for one of these queries which contains a word not observed in the collection are zero, unless some of the probability mass is allocated to words not seen in the corpus.

7.2 Lexical affinities

In (Nie & Dufort, 2002), compound terms have been incorporated as additional indices in the vector space model. Incorporating pairs can improve retrieval performance, but this depends largely on how the pairs are incorporated. Our lexical affinity model AL-1, based primarily on pairs, with single words included for smoothing (section 5.2.1) does not outperform the vector model or the unigram and bigram models. This is true regardless of the score function, as the probability of generating the query and the KL-divergence give similar results. This is probably due to the relative contributions of pairs and single words in this model. Regardless of the value for the weighting coefficient of observed pairs, λ_2, the contribution of single terms in the relevance score is significantly lower than the contribution of pairs. If we consider an example document that contains 150 unique words and 200 unique word pairs, the average probability of a single word is $1/150 = 0.0067$. The total number of possible pairs is $150 \times 150 = 22500$, and the number of these pairs that do not exist in the document is therefore $22500 - 200 = 22300$. The value of γ in equation 17, for this particular document, is then:

$$\gamma = \frac{1 - \lambda_2}{22300 \times (0.0067 + 0.0067)} = \frac{1 - \lambda_2}{297} \tag{31}$$

We can see that the contribution of pairs in equation 17 will be disproportionately large relative to the single words.

Our model AL-2, which combines single words with pairs represented as single terms (section 5.3) outperforms the best unigram and bigram models, for certain relative contributions of pairs. If the relative contribution of pairs, in the query or in the document, is too large however, the performance decreases. The optimal contribution of pairs is on the order of 0.1 times the contribution of single terms.

The gain in effectiveness obtained by the model AL-2, compared to a unigram model that only

contains single words, is greater when a fixed smoothing $p_d(unk)$ is used in both cases, than when a corpus model is used for smoothing (also in both cases). In addition, the increase in performance with model AL-2 is greater for the shorter queries than for the longer queries, whether the smoothing is fixed or involves a corpus model. The title queries are typically 2-3 words long, and contain 2-3 lexical affinities, whereas the title and description queries are about 6-7 words long on average, and contain an average of 18 pairs. It is possible that the longer queries contain some pairs that are not completely relevant resulting in noise in the relevance score. Filtering the pairs in the queries may provide some increase in performance.

8 Conclusion and future work

We have observed an improvement in retrieval performance when word pairs are included in the language modeling approach to information retrieval. However, several improvements may be possible. In our experiments involving lexical affinities, we have included all word pairs within a window of 5 words (to the left and right of a given word), with filtering based on the resolving power of the pairs. It may be possible to improve performance with a parser, using only the pairs in a particular syntactical relationship such as noun phrases (i.e., "computer science") or words in a modifier-modified relationship (i.e., "prime minister"). The size of the window from which words are extracted is another parameter that may be tested to improve performance. Filtering the pairs in the queries, by a parser, or by resolving power, would most likely allow a greater performance increase for the longer queries than for the shorter queries. Another improvement may be possible by using an EM algorithm to automatically learn parameters such as β_d and β_q, per document, rather than determining them empirically and fixed for the set of documents. Finally, there may be other ways to integrate word pairs within language models. One possible approach could involve training two language models for a document: one that contains the single words, and the other that contains the pairs. The probability of generating the query would be calculated for both models, and the relevance score would be a combination these two probabilities.

References

Berger, A. & Lafferty, J. (1999). Information retrieval as statistical translation. In *Research and development in information retrieval*, (pp. 222–229).

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal*, **35**(3), 243–255.

Hiemstra, D. (2002). Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *25th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, (pp. 35–41), Tampere, Finland.

Hiemstra, D. & de Vries, A. (2000). *Relating the new language models of information retrieval to the traditional retrieval models.* Rapport interne Technical report TR-CTIT-00-09, Centre for Telematics and Information Technology.

Jiang, M., Jensen, E. & Bietzel, S. (2004). Effective use of phrases in language modeling to improve information retrieval. In *Eighth international symposium on artificial intelligence and mathematics*, (.

Jin, R., Hauptmann, A. & Zhai, C. (2002). Title language model for information retrieval. In *Proceedings on the 25th annual international ACM SIGIR conference*, (pp. 42–48).

Lafferty, J. & Berger, A. (1999). The weaver system for document retrieval. In *Proceedings of the eighth Text REtrieval Conference (TREC-8)*, (.

Lafferty, J. & Zhai, J. (2001). Document language models, query models and risk minimization for information retrieval. In *24th annual international ACM SIGIR conference*, (pp. 111–119), New Orleans, Louisiana.

Lavrenko, V. & Croft, W. B. (2001). Relevance-based language models. In *24th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, (pp. 120–127).

Maarek, Y., Berry, D. & Kaiser, G. (1991). An information retrieval approach for automatically constructing software libraries. *IEEE transactions on software engineering*, (pp. 800–813).

Martin, W., Al, B. & Sterkenburg, P. V. (1983). On the processing of a text corpus: From textual data to lexicographical information. In E. R.R.K. Hartmann, Ed., *Lexicography: Principles and Practice*, Applied Language Studies Series, (. Academic Press, London.

Nie, J.-Y. & Dufort, J. (2002). Combining words and compound terms for monolingual and cross-language information retrieval. In *Information 2002*, (.

Pickens, J. (2000). A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval. In *Proceedings of the first international symposium for music information retrieval (ISMIR)*, (.

Ponte, J. M. & Croft, W. B. (1998). A language modeling approach to information retrieval. In *21st annual international ACM SIGIR conference on Research and Development in Information Retrieval*, (pp. 275–281), Melbourne, Australia.

Song, F. & Croft, W. B. (1999). A general language model for information retrieval. In *22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, (pp. 279–280).

Srikanth, M. & Srihari, R. (2002). Biterm language models for document retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, (pp. 425–426), Tampere, Finland.