Language Model Information Retrieval with Document Expansion

Tao Tao, Xuanhui Wang, Qiaozhu Mei, ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana_Champaign

Abstract

Language model information retrieval depends on accurate estimation of document models. In this paper, we propose a document expansion technique to deal with the problem of insufficient sampling of documents. We construct a probabilistic neighborhood for each document, and expand the document with its neighborhood information. The expanded document provides a more accurate estimation of the document model, thus improves retrieval accuracy. Moreover, since document expansion and pseudo feedback exploit different corpus structures, they can be combined to further improve performance. The experiment results on several different data sets demonstrate the effectiveness of the proposed document expansion method.

1 Introduction

Information retrieval with statistical language models (Lafferty and Zhai, 2003) has recently attracted much more attention because of its solid theoretical background as well as its good empirical performance. In this approach, queries and documents are assumed to be sampled from hidden generative models, and the similarity between a document and a query is then calculated through the similarity between their underlying models.

Clearly, good retrieval performance relies on the accurate estimation of the query and document models. Indeed, smoothing of document models has been proved to be very critical (Chen and Goodman, 1998; Kneser and Ney, 1995; Zhai and Lafferty, 2001b). The need for smoothing originated from the zero count problem: when a term does not occur in a document, the maximum likelihood estimator would give it a zero probability. This is unreasonable because the zero count is often due to insufficient sampling, and a larger sample of the data would likely contain the term. Smoothing is proposed to address the problem.

While most smoothing methods utilize the global collection information with a simple interpolation (Ponte and Croft, 1998; Miller et al., 1999; Hiemstra and Kraaij, 1998; Zhai and Lafferty, 2001b), several recent studies (Liu and Croft, 2004; Kurland and Lee, 2004) have shown that local corpus structures can be exploited to improve retrieval performance. In this paper, we further study the use of local corpus structures for document model estimation and propose to use document expansion to better exploit local corpus structures for estimating document language models.

According to statistical principles, the accuracy of a statistical estimator is largely determined by the sampling size of the observed data; a small data set generally would result in large variances, thus can not be trusted completely. Unfortunately, in retrieval, we often have to estimate a model based on a single document. Since a document is a small sample, our estimate is unlikely to be very accurate.

A natural improvement is to enlarge the data sample, ideally in a document-specific way. Ideally, the enlarged data sample should come from the same original generative model. In reality, however, since

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 407–414,
New York, June 2006. ©2006 Association for Computational Linguistics

the underlying model is unknown to us, we would not really be able to obtain such extra data. The essence of this paper is to use *document expansion* to obtain high quality extra data to enlarge the sample of a document so as to improve the accuracy of the estimated document language model. Document expansion was previously explored in (Singhal and Pereira, 1999) in the context of the vector space retrieval model, mainly involving selecting more terms from similar documents. Our work differs from this previous work in that we study document expansion in the language modeling framework and implement the idea quite differently.

Our main idea is to augment a document probabilistically with potentially all other documents in the collection that are similar to the document. The probability associated with each neighbor document reflects how likely the neighbor document is from the underlying distribution of the original document, thus we have a "probabilistic neighborhood", which can serve as "extra data" for the document for estimating the underlying language model. From the viewpoint of smoothing, our method extends the existing work on using clusters for smoothing (Liu and Croft, 2004) to allow each document to have its own cluster for smoothing.

We evaluated our method using six representative retrieval test sets. The experiment results show that document expansion smoothing consistently outperforms the baseline smoothing methods in all the data sets. It also outperforms a state-of-the-art clustering smoothing method. Analysis shows that the improvement tends to be more significant for short documents, indicating that the improvement indeed comes from the improved estimation of the document language model, since a short document presumably would benefit more from the neighborhood smoothing. Moreover, since document expansion and pseudo feedback exploit different corpus structures, they can be combined to further improve performance. As document expansion can be done in the indexing stage, it is scalable to large collections.

2 Document Expansion Retrieval Model

2.1 The KL-divergence retrieval model

We first briefly review the KL-divergence retrieval model, on which we will develop the document expansion technique. The KL-divergence model is a representative state-of-the-art language modeling approach for retrieval. It covers the basic language modeling approach (i.e., the query likelihood method) as a special case and can support feedback more naturally.

In this approach, a query and a document are assumed to be generated from a unigram query language model Θ_Q and a unigram document language model Θ_D, respectively. Given a query and a document, we would first compute an estimate of the corresponding query model ($\hat{\Theta}_Q$) and document model ($\hat{\Theta}_D$), and then score the document w.r.t. the query based on the KL-divergence of the two models (Lafferty and Zhai, 2001):

$$D(\hat{\Theta}_Q \mid\mid \hat{\Theta}_d) = \sum_{w \in V} p(w|\hat{\Theta}_Q) \times \log \frac{p(w|\hat{\Theta}_Q)}{p(w|\hat{\Theta}_d)}.$$

where V is the set of all the words in our vocabulary. The documents can then be ranked according to the ascending order of the KL-divergence values.

Clearly, the two fundamental problems in such a model are to estimate the query model and the document model, and the accuracy of our estimation of these models would affect the retrieval performance significantly. The estimation of the query model can often be improved by exploiting the local corpus structure in a way similar to pseudo-relevance feedback (Lafferty and Zhai, 2001; Lavrenko and Croft, 2001; Zhai and Lafferty, 2001a). The estimation of the document model is most often done through smoothing with the global collection language model (Zhai and Lafferty, 2001b), though recently there has been some work on using clusters for smoothing (Liu and Croft, 2004). Our work is mainly to extend the previous work on document smoothing and improve the accuracy of estimation by better exploiting the local corpus structure. We now discuss all these in detail.

2.2 Smoothing of document models

Given a document d, the simplest way to estimate the document language model is to treat the document as a sample from the underlying multinomial word distribution and use the maximum likelihood estimator: $P(w|\hat{\Theta}_d) = \frac{c(w,d)}{|d|}$, where $c(w,d)$ is the count of word w in document d, and $|d|$ is the length of d. However, as discussed in virtually all the existing work on using language models for retrieval, such an estimate is problematic and inaccurate; indeed, it would assign zero probability to any word not present in document d, causing problems in scoring a document with query likelihood or KL-divergence (Zhai and Lafferty, 2001b). Intuitively, such an estimate is inaccurate because the document is a small sample.

To solve this problem, many different smoothing techniques have been proposed and studied, usually involving some kind of interpolation of the maximum likelihood estimate and a global collection language model (Hiemstra and Kraaij, 1998; Miller et al., 1999; Zhai and Lafferty, 2001b). For example, Jelinek-Mercer(JM) and Dirichlet are two commonly used smoothing methods (Zhai and Lafferty, 2001b). JM smoothing uses a fixed parameter λ to control the interpolation:

$$P(w|\hat{\Theta}_d) = \lambda \frac{c(w,d)}{|d|} + (1-\lambda) P(w|\Theta_C),$$

while the Dirichlet smoothing uses a document-dependent coefficient (parameterized with μ) to control the interpolation:

$$P(w|\hat{\Theta}_d) = \frac{c(w,d) + \mu P(w|\Theta_C)}{|d| + \mu}.$$

Here $P(w|\Theta_C)$ is the probability of word w given by the collection language model Θ_C, which is usually estimated using the whole collection of documents C, e.g., $P(w|\Theta_C) = \frac{\sum_{d \in C} c(d,w)}{\sum_{d \in C} |d|}$.

2.3 Cluster-based document model (CBDM)

Recently, the cluster structure of the corpus has been exploited to improve language models for retrieval (Kurland and Lee, 2004; Liu and Croft, 2004). In particular, the cluster-based language model proposed in (Liu and Croft, 2004) uses clustering information to further smooth a document model. It divides all documents into K different clusters ($K = 1000$ in their experiments). Both cluster information and collection information are used to improve the estimate of the document model:

$$\begin{aligned} P(w|\hat{\Theta}_d) &= \lambda \frac{c(w,d)}{|d|} + (1-\lambda) \\ &\quad \times [\beta P(w|\Theta_{L_d}) + (1-\beta) P(w|\Theta_C)], \end{aligned}$$

where Θ_{L_d} stands for document d's cluster model and λ and β are smoothing parameters. In this clustering-based smoothing method, we first smooth a cluster model with the collection model using Dirichlet smoothing, and then use smoothed cluster model as a new reference model to further smooth the document model using JM smoothing; empirical results show that the added cluster information indeed enhances retrieval performance (Liu and Croft, 2004).

2.4 Document expansion

From the viewpoint of data augmentation, the clustering-based language model can be regarded as "expanding" a document with more data from the cluster that contains the document. This is intuitively better than simply expanding every document with the same collection language model as in the case of JM or Dirichlet smoothing. Looking at it from this perspective, we see that, as the "extra data" for smoothing a document model, the cluster containing the document is often not optimal. Indeed, the purpose of clustering is to group similar documents together, hence a cluster model represents well the overall property of *all* the documents in the cluster. However, such an average model is often not accurate for smoothing each individual document. We illustrate this problem in Figure 1(a), where we show two documents d and a in cluster D. Clearly the generative model of cluster D is more suitable for smoothing document a than document d. In general, the cluster model is more suitable for smoothing documents close to the centroid, such as a, but is inaccurate for smoothing a document at the boundary, such as d.

To achieve optimal smoothing, each document should ideally have its own cluster centered on the document, as shown in Figure 1(b). This is precisely what we propose – expanding each document with a probabilistic neighborhood around the document and estimate the document model based on such a virtual, expanded document. We can then apply any simple interpolation-based method (e.g., JM or Dirichlet) to such a "virtual document" and treat the word counts given by this "virtual document" as if they were the original word counts.

The use of neighborhood information is worth more discussion. First of all, neighborhood is not a

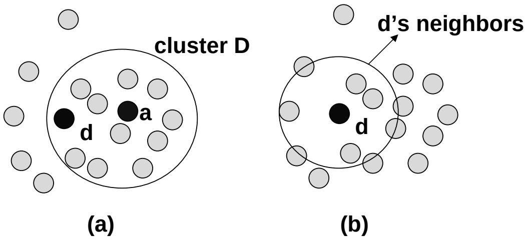

Figure 1: Clusters, neighborhood, and document expansion

clearly defined concept. In the narrow sense, only a few documents close to the original one should be included in the neighborhood, while in the wide sense, the whole collection can be potentially included. It is thus a challenge to define the neighborhood concept reasonably. Secondly, the assumption that neighbor documents are sampled from the same generative model as the original document is not completely valid. We probably do not want to trust them so much as the original one. We solve these two problems by associating a confidence value with every document in the collection, which reflects our belief that the document is sampled from the same underlying model as the original document. When a document is close to the original one, we have high confidence, but when it is farther apart, our confidence would fade away. In this way, we construct a probabilistic neighborhood which can potentially include all the documents with different confidence values. We call a language model based on such a neighborhood *document expansion language model* (DELM).

Technically, we are looking for a new *enlarged* document d' for each document d in a text collection, such that the new document d' can be used to estimate the hidden generative model of d more accurately. Since a good d' should presumably be based on both the original document d and its neighborhood $N(d)$, we define a function ϕ:

$$d' = \phi(d, N(d)). \tag{1}$$

The precise definition of the neighborhood concept $N(d)$ relies on the distance or similarity between each pair of documents. Here, we simply choose the commonly used cosine similarity, though other choices may also be possible. Given any two document models X and Y, the cosine similarity is defined as:

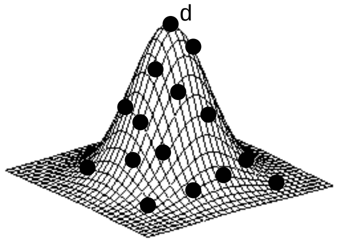

Figure 2: Normal distribution of confidence values.

$$sim(X,Y) = \frac{\sum_i x_i \times y_i}{\sqrt{\sum_i (x_i)^2 \times \sum_i (y_i)^2}}.$$

To model the uncertainty of neighborhood, we assign a confidence value $\gamma_d(b)$ to every document b in the collection to indicate how strongly we believe b is sampled from d's hidden model. In general, $\gamma_d(b)$ can be set based on the similarity of b and d – the more similar b and d are, the larger $\gamma_d(b)$ would be. With these confidence values, we construct a probabilistic neighborhood with every document in it, each with a different weight. The whole problem is thus reduced to how to define $\gamma_d(b)$ exactly.

Intuitively, an exponential decay curve can help regularize the influence from remote documents. We therefore want $\gamma_d(b)$ to satisfy a normal distribution centered around d. Figure 2 illustrates the shape of this distribution. The black dots are neighborhood documents centered around d. Their probability values are determined by their distances to the center. We fortunately observe that the cosine similarities, which we use to decide the neighborhood, are roughly of this decay shape. We thus use them directly without further transformation because that would introduce unnecessary parameters. We set $\gamma_d(b)$ by normalizing the cosine similarity scores :

$$\gamma_d(b) = \frac{sim(d,b)}{\sum_{b' \in C - \{d\}} sim(d,b')}.$$

Function ϕ serves to balance the confidence between d and its neighborhood $N(d)$ in the model estimation step. Intuitively, a shorter document is less sufficient, hence needs more help from its neighborhood. Conversely, a longer one can rely more on itself. We use a parameter α to control this balance. Thus finally, we obtain a pseudo document d' with the following pseudo term count:

$$c(w,d') = \alpha c(w,d) + (1-\alpha) \times \sum_{b \in C-\{d\}} (\gamma_d(b) \times c(w,b)),$$

We hypothesize that, in general, Θ_d can be estimated more accurately from d' rather than d itself because d' contains more complete information about Θ_d. This hypothesis can be tested by by comparing the retrieval results of applying any smoothing method to d with those of applying the same method to d'. In our experiments, we will test this hypothesis with both JM smoothing and Dirichlet smoothing.

Note that the proposed document expansion technique is quite general. Indeed, since it transforms the original document to a potentially better "expanded document", it can presumably be used together with any retrieval method, including the vector space model. In this paper, we focus on evaluating this technique with the language modeling approach.

Because of the decay shape of the neighborhood and for the sake of efficiency, we do not have to actually use all documents in $C - \{d\}$. Instead, we can safely cut off the documents on the tail, and only use the top M closest neighbors for each document. We show in the experiment section that the performance is not sensitive to the choice of M when M is sufficiently large (for example 100). Also, since document expansion can be done completely offline, it can scale up to large collections.

3 Experiments

We evaluate the proposed method over six representative TREC data sets (Voorhees and Harman, 2001): AP (Associated Press news 1988-90), LA (LA Times), WSJ (Wall Street Journal 1987-92), SJMN (San Jose Mercury News 1991), DOE (Department of Energy), and TREC8 (the ad hoc data used in TREC8). Table 1 shows the statistics of these data.

We choose the first four TREC data sets for performance comparison with (Liu and Croft, 2004). To ensure that the comparison is meaningful, we use identical sources (after all preprocessing). In addition, we use the large data set TREC8 to show that our algorithm can scale up, and use DOE because its documents are usually short, and our previous experience shows that it is a relatively difficult data set.

	#document	queries	#total qrel
AP	242918	51-150	21819
LA	131896	301-400	2350
WSJ	173252	51-100 and 151-200	10141
SJMN	90257	51-150	4881
TREC8	528155	401-450	4728
DOE	226087	DOE queries	2047

Table 1: Experiment data sets

3.1 Neighborhood document expansion

Our model boils down to a standard query likelihood model when no neighborhood document is used. We therefore use two most commonly used smoothing methods, JM and Dirichlet , as our baselines. The results are shown in Table 2, where we report both the mean average precision (MAP) and precision at 10 documents. JM and Dirichlet indicate the standard language models with JM smoothing and Dirichlet smoothing respectively, and the other two are the ones combined with our document expansion. For both baselines, we tune the parameters (λ for JM, and μ for Dirichlet) to be optimal. We then use the same values of λ or μ without further tuning for the document expansion runs, which means that the parameters may not necessarily optimal for the document expansion runs. Despite this disadvantage, we see that the document expansion runs significantly outperform their corresponding baselines, with more than 15% relative improvement on AP. The parameters M and α were set to 100 and 0.5, respectively.

To understand the improvement in more detail, we show the precision values at different levels of recall for the AP data in Table 3. Here we see that our method significantly outperforms the baseline at every precision point.

In our model, we introduce two additional parameters: M and α. We first examine M here, and then study α in Section 3.3. Figure 3 shows the performance trend with respect to the values of M. The x-axis is the values of M, and the y-axis is the non-interpolated precision averaging over all 50 queries. We draw two conclusions from this plot: (1) Neighborhood information improves retrieval accuracy; adding more documents leads to better retrieval results. (2) The performance becomes insensitive to

Data		JM	DELM+JM (impr. %)		Dirichlet	DELM + Diri.(impr. %)	
AP	AvgPrec	0.2058	0.2405	(16.8%***)	0.2168	0.2505	(15.5%***)
	P@10	0.3990	0.4444	(11.4%***)	0.4323	0.4515	(4.4%**)
DOE	AvgPrec	0.1759	0.1904	(8.3%***)	0.1804	0.1898	(5.2%**)
	P@10	0.2629	0.2943	(11.9%*)	0.2600	0.2800	(7.7%*)
TREC8	AvgPrec	0.2392	0.2539	(6.01%**)	0.2567	0.2671	(4.05%*)
	P@10	0.4300	0.4460	(3.7%)	0.4500	0.4740	(5.3%*)

Table 2: Comparisons with baselines. *,**,*** indicate that we accept the improvement hypothesis by Wilcoxon test at significance level 0.1, 0.05, 0.01 respectively.

AP, TREC queries 51-150			
	Dirichlet	DELM+Diri	Improvement(%)
Rel.	21819	21819	
Rel.Retr.	10126	10917	7.81% ***
Prec.			
0.0	0.6404	0.6605	3.14% *
0.1	0.4333	0.4785	10.4% ***
0.2	0.3461	0.3983	15.1% ***
0.3	0.2960	0.3496	18.1% ***
0.4	0.2436	0.2962	21.6% ***
0.5	0.2060	0.2418	17.4% ***
0.6	0.1681	0.1975	17.5% ***
0.7	0.1290	0.1580	22.5% ***
0.8	0.0862	0.1095	27.0% **
0.9	0.0475	0.0695	46.3% **
1.0	0.0220	0.0257	16.8%
ave.	0.2168	0.2505	15.5% ***

Table 3: PR curve on AP data. *,**,*** indicate that we accept the improvement hypothesis by Wilcoxon test at significant level 0.1, 0.05, 0.01 respectively.

M when M is sufficiently large, namely 100. The reason is twofold: First, since the neighborhood is centered around the original document, when M is large, the expansion may be evenly magnified on all term dimensions. Second, the exponentially decaying confidence values reduce the influence of remote documents.

3.2 Comparison with CBDM

In this section, we compare the CBDM method using the model performing the best in (Liu and Croft, 2004)[1]. Furthermore, we also set Dirichlet prior parameter $\mu = 1000$, as mentioned in (Liu and Croft, 2004), to rule out any potential influence of Dirichlet smoothing.

Table 4 shows that our model outperforms CBDM in MAP values on four data sets; the improvement presumably comes from a more principled way of exploiting corpus structures. Given that clustering can at least capture the local structure to some extent, it should not be very surprising that the improvement of document expansion over CBDM is much less than that over the baselines.

Note that we cannot fulfill Wilcoxon test because of the lack of the individual query results of CBDM.

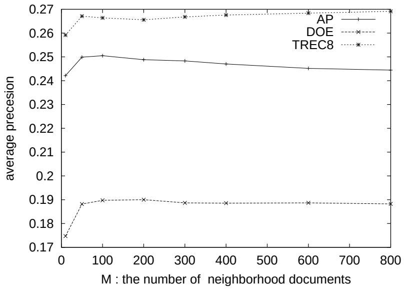

Figure 3: Performance change with respect to M

	CBDM	DELM+Diri.	improvement(%)
AP	0.2326	0.2505	7.7%
LA	0.2590	0.2655	2.5%
WSJ	0.3006	0.3113	3.6%
SJMN	0.2171	0.2266	4.3%

Table 4: Comparisons with CBDM.

3.3 Impact on short documents

Document expansion is to solve the insufficient sampling problem. Intuitively, a short document is *less* sufficient than a longer one, hence would need more "help" from its neighborhood. We design experiments to test this hypothesis.

Specifically, we randomly shrink each document in AP88-89 to a certain percentage of its original length. For example, a shrinkage factor of 30% means each term has 30% chance to stay, or 70% chance to be filtered out. In this way, we reduce the original data set to a new one with the same number

[1] We use the exact same data, queries, stemming and all other preprocessing techniques. The baseline results in (Liu and Croft, 2004) are confirmed.

average doc length	30%	50%	70%	100%
baseline	0.1273	0.1672	0.1916	0.2168
document expansion	0.1794	0.2137	0.2307	0.2505
optimal α	0.2	0.3	0.3	0.4
improvement(%)	41%	28%	20%	16%

Table 5: Impact on short documents (in MAP)

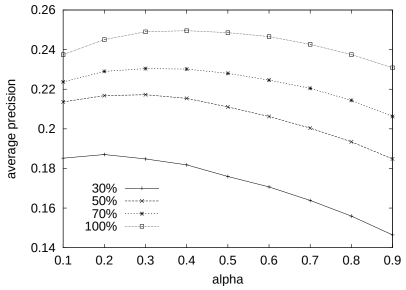

Figure 4: Performance change with respect to α

	DELM	pseudo	DELM+pseudo	Impr.(%)
AP	0.2505	0.2643	0.2726	3.14%*
LA	0.2655	0.2769	0.2901	4.77%
TREC8	0.2671	0.2716	0.2809	3.42%**
DOE	0.1898	0.1918	0.2046	6.67%***

Table 6: Combination with pseudo feedback.*,**,*** indicate that we accept the improvement hypothesis by Wilcoxon test at significant level 0.1, 0.05, 0.01 respectively.

	pseu.	inter.	combined (%)	z-score
AP	0.2643	0.2450	0.2660 (0.64%)	-0.2888
LA	0.2769	0.2662	0.2636 (-0.48%)	-1.0570
TREC8	0.2716	0.2702	0.2739 (0.84%)	-1.6938

Table 7: Performance of the interpolation algorithm combined with the pseudo feedback.

of documents but a shorter average document length.

Table 5 shows the experiment results over document sets with different average document lengths. The results indeed support our hypothesis that document expansion does help short documents more than longer ones. While we can manage to improve 41% on a 30%-length corpus, the same model only gets 16% improvement on the full length corpus.

To understand how α affects the performance we plot the sensitivity curves in Figure 4. The curves all look similar, but the optimal points slightly migrate when the average document length becomes shorter. A 100% corpus gets optimal at $\alpha = 0.4$, but 30% corpus has to use $\alpha = 0.2$ to obtain its optimum. (All optimal α values are presented in the fourth row of Table 5.)

3.4 Further improvement with pseudo feedback

Query expansion has been proved to be an effective way of utilizing corpus information to improve the query representation (Rocchio, 1971; Zhai and Lafferty, 2001a). It is thus interesting to examine whether our model can be combined with query expansion to further improve the retrieval accuracy. We use the model-based feedback proposed in (Zhai and Lafferty, 2001a) and take top 5 returned documents for feedback. There are two parameters in the model-based pseudo feedback process: the noisy parameter ρ and the interpolation parameter σ[2]. We fix $\rho = 0.9$ and tune σ to optimal, and use them directly in the feedback process combined with our models. (It again means that σ is probably not optimal in our results.) The combination is conducted in the following way: (1) Retrieve documents by our DELM method; (2) Choose top 5 document to do the model-based feedback; (3) Use the expanded query model to retrieve documents again with DELM method.

Table 6 shows the experiment results (MAP); indeed, by combining DELM with pseudo feedback, we can obtain significant further improvement of performance.

As another baseline, we also tested the algorithm proposed in (Kurland and Lee, 2004). Since the algorithm overlaps with pseudo feedback process, it is not easy to further combine them. We implement its best-performing algorithm, “interpolation” (labeled as inter.), and show the results in Table 7. Here, we use the same three data sets as used in (Kurland and Lee, 2004). We tune the feedback parameters to optimal in each experiment. The second last column in Table 7 shows the performance of combination of the “interpolation” model with the pseudo feedback and its improvement percentage. The last column is the z-scores of Wilcoxon test. The negative z-scores indicate that none of the improvement is significant.

[2] (Zhai and Lafferty, 2001a) uses different notations. We change them because α has already been used in our own model.

4 Conclusions

In this paper, we proposed a novel document expansion method to enrich the document sample through exploiting the local corpus structure. Unlike previous cluster-based models, we smooth each document using a probabilistic neighborhood centered around the document itself.

Experiment results show that (1) The proposed document expansion method outperforms both the "no expansion" baselines and the cluster-based models. (2) Our model is relatively insensitive to the setting of parameter M as long as it is sufficiently large, while the parameter α should be set according to the document length; short documents need a smaller α to obtain more help from its neighborhood. (3) Document expansion can be combined with pseudo feedback to further improve performance. Since any retrieval model can be presumably applied on top of the expanded documents, we believe that the proposed technique can be potentially useful for any retrieval model.

5 Acknowledgments

This work is in part supported by the National Science Foundation under award number IIS-0347933. We thank Xiaoyong Liu for kindly providing us several processed data sets for our performance comparison. We thank Jing Jiang and Azadeh Shakery for helping improve the paper writing, and thank the anonymous reviewers for their useful comments.

References

S. F. Chen and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University.

D. Hiemstra and W. Kraaij. 1998. Twenty-one at trec-7: Ad-hoc and cross-language track. In *Proc. of Seventh Text REtrieval Conference (TREC-7).*

R. Kneser and H. Ney. 1995. Improved smoothing for m-gram languagemodeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*.

Oren Kurland and Lillian Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 194–201. ACM Press.

John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR'2001*, pages 111–119, Sept.

John Lafferty and ChengXiang Zhai. 2003. Probabilistic relevance models based on document and query generation.

Victor Lavrenko and Bruce Croft. 2001. Relevance-based language models. In *Proceedings of SIGIR'2001*, Sept.

Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 186–193. ACM Press.

D. H. Miller, T. Leek, and R. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214–221.

J. Ponte and W. B. Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR*, pages 275–281.

J. Rocchio. 1971. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall Inc.

Amit Singhal and Fernando Pereira. 1999. Document expansion for speech retrieval. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41. ACM Press.

E. Voorhees and D. Harman, editors. 2001. *Proceedings of Text REtrieval Conference (TREC1-9).* NIST Special Publications. http://trec.nist.gov/pubs.html.

Chengxiang Zhai and John Lafferty. 2001a. Model-based feedback in the KL-divergence retrieval model. In *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 403–410.

Chengxiang Zhai and John Lafferty. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'2001*, pages 334–342, Sept.