

Language-Modeling Kernel Based Approach for Information Retrieval

Ying Xie

*Department of Computer Science and Information Systems, Kennesaw State University, Kennesaw, GA 30144.
E-mail: yxie2@kennesaw.edu*

Vijay V. Raghavan

*Center for Advanced Computer Studies, University of Louisiana, Lafayette, LA 70504.
E-mail: raghavan@cacs.louisiana.edu*

In this presentation, we propose a novel integrated information retrieval approach that provides a unified solution for two challenging problems in the field of information retrieval. The first problem is how to build an optimal vector space corresponding to users' different information needs when applying the vector space model. The second one is how to smoothly incorporate the advantages of machine learning techniques into the language modeling approach. To solve these problems, we designed the language-modeling kernel function, which has all the modeling powers provided by language modeling techniques. In addition, for each information need, this kernel function automatically determines an optimal vector space, for which a discriminative learning machine, such as the support vector machine, can be applied to find an optimal decision boundary between relevant and nonrelevant documents. Large-scale experiments on standard test-beds show that our approach makes significant improvements over other state-of-the-art information retrieval methods.

Introduction

The vector space model (VSM) is by far the most widely utilized computational model for information retrieval. Most Web search engines have adopted similar strategies. The assumption of applying the VSM is that we can construct a vector space, so that documents can be properly represented as vectors in that space. If this assumption holds, the VSM provides an elegant way to model similarity relationship among documents, which enables machine learning algorithms such as support vector machines (SVM) to construct optimal decision boundaries based on the relevance feedback to achieve better retrieval performance. Therefore,

the key issues in applying the VSM for information retrieval are (a) finding an optimal vector space and (b) properly representing documents in that vector space; however, to our knowledge, there is no systematic way to solve these two issues. Both the construction of the vector space and the representation of documents in the vector space are conducted heuristically. Furthermore, we found that the greatest challenge in applying the VSM is that the similarity relationship among documents varies from one information need to another. In other words, different information needs will result in different vector spaces. This phenomenon can be illustrated by a simple example. Assume we have a document collection with three documents:

- D1: Recommended computer science curriculum changes.
- D2: Integrating computer ethics into the computer science curriculum.
- D3: Essays on ethics and misconduct in science.

Given the query “computer science curriculum,” the similarity between D1 and D2 is obviously greater than any other pair in this collection; for the query “science ethics,” the similarity between D2 and D3 is the greatest. Therefore, when a machine learning algorithm is applied on the training data to find a decision boundary that separates relevant documents from nonrelevant ones, the query specific distance between documents should perform better than the absolute distance.

Proposed recently as an alternative paradigm to traditional methods for information retrieval, the language-modeling approach models and estimates term distribution for each document, and ranks documents based on the probability that a query would be generated through a random sampling process from the respective term distributions (Ponté & Croft, 1998). This new approach not only provides a well-interpreted way to utilize document statistics but also outperforms the basic VSM with a term frequency-inverse document frequency (TF-IDF) indexing scheme on several

Received June 3, 2006; revised April 2, 2007, April 22, 2007; accepted June 17, 2007

© 2007 Wiley Periodicals, Inc. • Published online 21 September 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20711

TREC collections; however, the lack of an explicit model for relevance makes it conceptually difficult to incorporate relevance feedback mechanisms into the language modeling approach. To overcome this obstacle, the *model based feedback approach* (Zhai & Lafferty, 2001a) utilizes relevant documents to refine the query language model (i.e., the term distribution from which the query is generated) and ranks the documents based on the divergences between the query language model and document language models. Although this work enables the language modeling approach to acquire limited learning ability, the problem of how to incorporate the advantages brought by machine learning algorithms, such as the SVM, into the language modeling technique to further enhance the retrieval performance remains unexplored.

Instead of viewing them as alternative paradigms, we propose that language modeling techniques and the VSM should be viewed as complementary. Language modeling techniques contribute a systematic way to estimate and utilize term distributions, which is lacking in the VSM. While the VSM provides an abstract interface to interact with machine learning algorithms, such as the SVM, which thus far has been impossible for the language modeling approach. Therefore, it is reasonable to expect better retrieval performance if we can integrate both language modeling techniques and the VSM into a unified framework. In this work, we propose such an integrated framework, called the *language-modeling kernel based approach*, for information retrieval.

The key component that integrates language modeling techniques and the VSM is called the *language-modeling kernel*. The language-modeling kernel is designed in such a way that it preserves all the modeling power provided by language modeling techniques. Moreover, this kernel function dynamically and systematically maps documents into a high dimensional vector space by taking advantage of document statistics (i.e., term-frequency information), collection statistics (i.e., term-term co-occurrence information), and relevance statistics. The SVM then can be applied in this mapped high dimensional vector space to find an optimal boundary that separates the relevant documents from the nonrelevant ones.

From the perspective of the VSM, the language-modeling kernel based approach resolves the problem of systematically and dynamically constructing optimal vector spaces that tailor to users' different information needs by the means of language modeling techniques; while from the perspective of the language modeling approach, the language-modeling kernel based approach is able to incorporate advantages of the SVM into language modeling techniques.

The rest of the article is organized as follows. First we briefly review the VSM and the language modeling approach. Based upon the reviews, we then present our contribution, the *language-modeling kernel based approach for information retrieval*. Experimental results that show significant improvements made by our approach over the state-of-the-art retrieval methods will be given next. Finally, the conclusion outlines our contributions and envisions future

work on personalized information retrieval based on our proposed language-modeling kernel.

Review of Major Information Retrieval Models

As is well known, the VSM views each document as a document vector and each user's query as a query vector. It ranks documents against the query based on the similarities between the query vector and document vectors; however, the VSM itself does not specify how to represent documents and query as vectors. Therefore, this model is counted as a ranking model, but not an indexing model (Ponte & Croft, 1998). To use the VSM to rank the documents against the query, some heuristics have to be used to build document vectors during the indexing process. A typical solution is to use a TF-IDF strategy to build document vectors. This strategy first builds a vector space by using all the unique terms appearing in the document collection as orthogonal dimensions. Then it uses term-frequency and document-frequency information to assign the weights of terms to each document.

By taking term-term co-occurrence information into consideration, the Generalized Vector Space Model (GVSM; Wong, Ziarko, & Wong, 1985) derives unique term combinations as orthogonal dimensions for the vector space, where each index term is represented as a vector according to its distribution over the document collections, while latent semantic indexing (LSI) utilizes the singular value decomposition (SVD) to derive a set of uncorrelated indexing factors as the orthogonal dimensions, which in fact are a group of linear combinations of documents (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990). Note that both the GVSM and the LSI start from the term-document matrix A , where the term weights have to be estimated in a heuristic way, such as the TF-IDF strategy.

Aiming at providing a systematic way to utilize term frequency for information retrieval, Ponte and Croft (1998) proposed applying language modeling techniques to information retrieval. The proposed language modeling approach estimates the document language model $P(w|M_D)$ for each Document D in a Collection C . By the maximum likelihood estimator (MLE), we have

$$\hat{P}_{mi}(w|M_D) = \frac{tf_{(w,D)}}{dl_D},$$

where $tf_{(w,D)}$ is the raw term frequency of Word w in Document D , and dl_D is the total number of tokens in D . Then, given a Query Q , Document D will be ranked based on the probability that this query string is observed by repeatedly sampling words from the document language model M_D . Given the term independence assumption, this ranking function can be simply expressed as:

$$P(Q|M_D) = \prod_{w \in Q} P(w|M_D). \quad (1)$$

To avoid assigning zero probability to a word that does not appear in Document D , various smoothing techniques

were proposed (Zhai & Lafferty, 2001b). A typical smoothing method called *linear interpolation smoothing*, which adjusts the maximum likelihood model with the collection model $P(w|C)$, whose influence is controlled by a coefficient parameter λ , can be expressed as follows:

$$P(Q|M_D) = \prod_{w \in Q} [\lambda p(w|M_D) + (1 - \lambda)P(w|C)].$$

Please refer to Zhai and Lafferty (2001b) for more details on smoothing techniques.

In addition to estimating term distribution for each document, Lafferty and Zhai (2001) expanded language modeling techniques to be able to utilize term-term co-occurrence information over the document corpus. They derived a Markov chain that alternates between words and documents from the index of this collection. By traversing this Markov chain, a word can be sampled from a document with the probability decided by the document language model. Once a word is sampled, a document can be selected from a list of documents including that word according to the posterior probability. Roughly speaking, the more documents that have Word A also include Word B, the higher probability that A can be translated into B.

To incorporate feedback mechanism into the language-modeling approach for information retrieval, Zhai and Lafferty (2001a) proposed a couple of *model-based feedback* techniques to estimate term distributions over relevant documents. The estimated term distribution for relevant documents is then used to refine the initial retrieval results.

The basic language modeling approach provides a well-interpreted way to utilize document statistics (i.e., term-frequency information). Unlike the VSM, which is just a retrieval model, the basic language modeling approach integrates the indexing and retrieval into the same model. The Markov chain expansion further enhances language modeling techniques with the capability of utilizing collection statistics (term-term co-occurrence information). Finally, the model-based feedback mechanism enables the modeling of relevance statistics.

However, it is conceptually difficult to incorporate the advantages of machine learning algorithms into language modeling techniques. Although the model-based feedback mechanism provides a way to incorporate relevance statistics from the positive samples into retrieval, this mechanism is unable to utilize statistics from the negative samples. The capability of learning from both positive samples and negative samples is a necessary condition for an effective learning method. Therefore, equipping language modeling techniques with the ability to learn from negative feedback is an important issue that needs to be addressed.

Language-Modeling Kernel Based Approach

After critically examining both the VSM and language modeling techniques for information retrieval, we find that these two types of models are essentially complementary. The VSM presents an abstract interface to interact with

machine learning algorithms; however, it does not provide a mechanism to systematically map given data into the vector space. On the contrary, language modeling techniques focus on the modeling of data, but it is conceptually difficult to integrate the current language modeling mechanism with the VSM such that all the advantages of machine learning algorithms are leveraged. This observation motivates us to design an integrated information retrieval framework in which documents can be systematically and dynamically represented as vectors based on document statistics, collection statistics, and relevance statistics by utilizing language modeling techniques. The architecture of this integrated framework for information retrieval is shown in Figure 1. The key component that integrates the VSM and language modeling techniques is a newly designed kernel function called the *language-modeling kernel*. The whole framework is called the *language-modeling kernel based approach for information retrieval*.

The language-modeling kernel based approach has the following features:

- It can systematically represent documents as vectors based on document statistics, collection statistics, and relevance statistics.
- It can dynamically determine a vector space based on the user's information need.
- It preserves all the modeling power provided by language modeling techniques.
- It can utilize both positive feedback and negative feedback.
- It can elegantly incorporate advantages of machine learning algorithms, such as the SVM, into language modeling techniques.

In other words, the language-modeling kernel based approach keeps all the advantages provided by both the VSM and language modeling techniques for information retrieval; meanwhile, it overcomes several challenging problems faced by these two types of models. Next, we introduce the language-modeling kernel, then describe the retrieval process based on the language-modeling kernel.

Language-Modeling Kernel

The language modeling approach estimates a document language model for each document in a given document collection. Therefore, the basic elements that the language modeling approach processes are document language models. The language-modeling kernel defined on document language models automatically maps document language models into a high dimensional vector space. To describe the way that the language-modeling kernel works, we first give a brief introduction to kernel functions.

Kernel functions. By Mercer's theory, every (semi)positive, definite, and symmetric function is a kernel. It determines a Map Φ , which maps the data from the input space to a high dimensional vector space, and the inner product in the

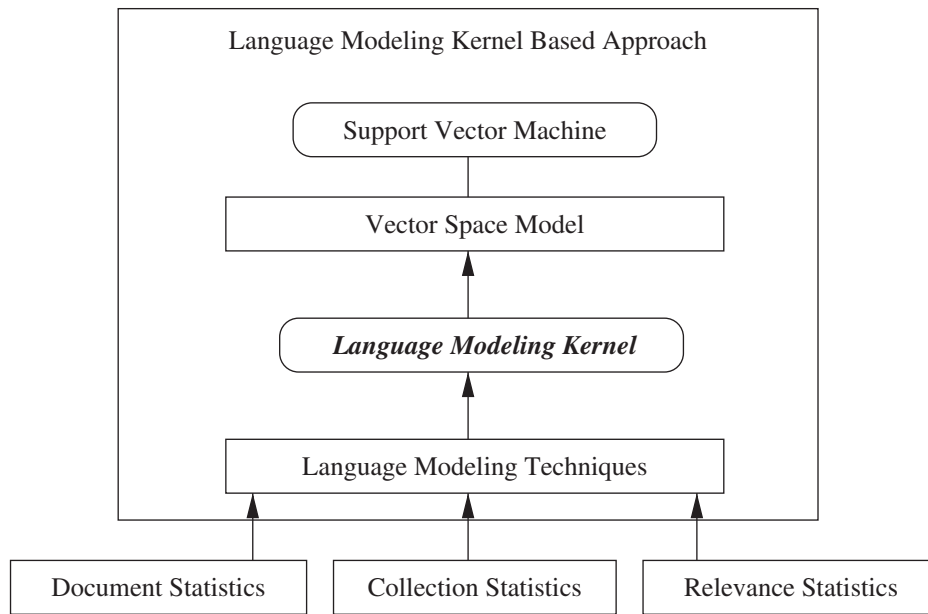


FIG. 1. The architecture of the language-modeling kernel based approach.

mapped vector space is equal to the kernel function in the original space. In formula, it can be expressed as:

$$K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$$

According to Mercer's theory, if we can define a kernel function on document language models, this kernel function can automatically map the document language models to a vector space. If we can define the kernel function on document language models in a way that it can take advantage of document statistics, collection statistics, and relevance statistics, then we find a systematic way to represent documents as vectors based upon document statistics, collection statistics, and relevance statistics. The following subsection describes the way we designed such a kernel function.

Design of the language-modeling kernel. Mercer's theory tells us that the inner dot product in the mapped vector space is equal to the kernel function in the original space. The inner dot product essentially evaluates a similarity relationship between two vectors. Therefore, the kernel function also should correspond to a similarity relationship between the data in the original input space. Hence, defining a proper measure to evaluate a similarity relationship between two document language models should lead to a way to define a proper kernel function for document language models. One of the challenging parts of the design is that we require the similarities among documents to vary along with the change in users' information needs.

Since a document language model describes certain probability distributions over the vocabulary provided by the given document collection, we begin our consideration with the existing similarity or distance measures for probability distributions. The KL-divergence measure is often utilized to evaluate the divergence between two probability distributions.

Given two document models M_{D1} and M_{D2} , the KL-divergence between M_{D1} and M_{D2} is defined as:

$$D(M_{D1} \| M_{D2}) = \sum_w P(w|M_{D1}) \log \frac{P(w|M_{D1})}{P(w|M_{D2})}.$$

Obviously, KL-divergence is not symmetric and does not satisfy the triangle inequality, therefore it cannot be directly applied to evaluate the distance between two document models. To overcome this problem, some symmetric variance of KL-divergence was proposed, for example:

$$D(M_{D1}, M_{D2}) = \sum_w P(w|M_{D1}) \log \frac{P(w|M_{D1})}{P(w|M_{D2})} + \sum_w P(w|M_{D2}) \log \frac{P(w|M_{D2})}{P(w|M_{D1})}.$$

However, the symmetric KL-divergence is still an absolute measure, whose evaluation is unable to change along with the change in users' information needs. To incorporate the user's information need into consideration, when designing the distance measure for document language models, we first have to find a way to model the user's information need. In Lafferty and Zhai (2001), a query is viewed as the result of choosing a language model (i.e., probability distribution of the vocabulary), and then generating the query using that model. Therefore, a user's particular information need can be modeled as a query language model $P(w|M_Q)$, from which the user's query is presumed to be randomly generated. Based on this view, we propose a distance measure for document language models as:

$$D(M_{D1}, M_{D2}) = \sum_w P(w|M_Q) \left| \log \frac{P(w|M_{D1})}{P(w|M_{D2})} \right|.$$

As can be seen, this distance measure is biased by the query language model that reflects the user's particular information need. By extension, we additionally introduce a query model biased similarity measure for document language models as follows:

$$K_{LM}(M_{D1}, M_{D2}) = e^{-A \sum_w P(w|M_q) \left| \log \frac{P(w|M_{D1})}{P(w|M_{D2})} \right| + B}, \quad (2)$$

where Parameter $A (>0)$ and B are scale and shift factors, respectively. This similarity measure has the following attributes: (a) $0 < K_{LM}(M_{D1}, M_{D2}) \leq e^B$; and (b) $K_{LM}(M_{D1}, M_{D2}) = e^B$, if and only if $M_{D1} = M_{D2}$. According to Mercer's theory, this positive, definite, and symmetric function is a kernel. We call it the *language-modeling kernel*. For simplicity, in this article, we set $A = 1$, and $B = 0$ for all the experiments. When $B = 0$, we have the following: (a) $0 < K_{LM}(M_{D1}, M_{D2}) \leq 1$; and (b) $K_{LM}(M_{D1}, M_{D2}) = 1$ if and only if $M_{D1} = M_{D2}$.

Unique features of the language-modeling kernel. Let us study the unique features of the language-modeling kernel. To estimate document language model components of this kernel,

one can utilize the simplest MLE. Even the simplest MLE takes advantages of the document statistics (i.e., term-frequency) information. More complex estimators, such as the Markov chain expansion (Lafferty & Zhai, 2001) mentioned in the last section, utilize not only the document statistics but also the collection statistics (i.e., term-term co-occurrence information). To estimate the query language model component, relevance statistics can be incorporated. Therefore, the language-modeling kernel preserves all the modeling power provided by language modeling techniques, as is illustrated in Figure 2.

The language-modeling kernel not only retains all the modeling power provided by language modeling techniques but also dynamically maps the document language models to different vector spaces according to users' different information needs. The query language model component of this kernel is used to model the user's information need. Different information needs will determine different language-modeling kernels. Different language-modeling kernels determine different high dimensional vector spaces. This process is illustrated in Figure 3.

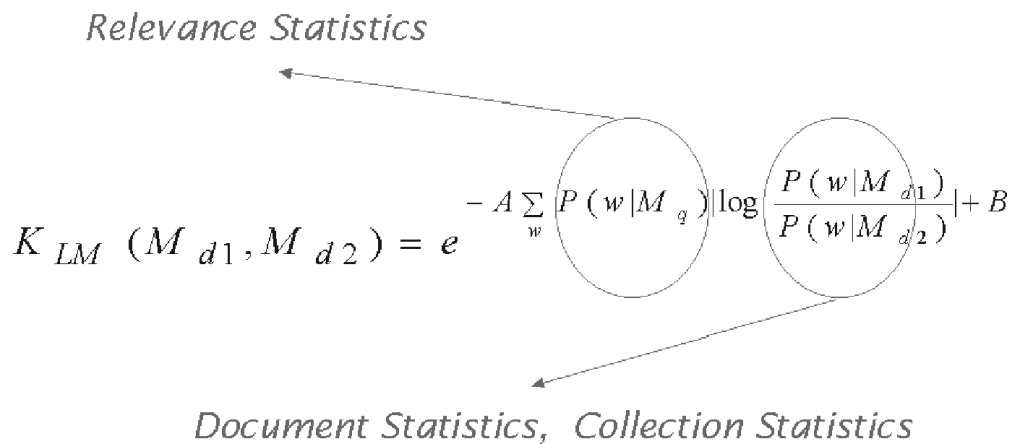


FIG. 2. The modeling power of the language-modeling kernel.

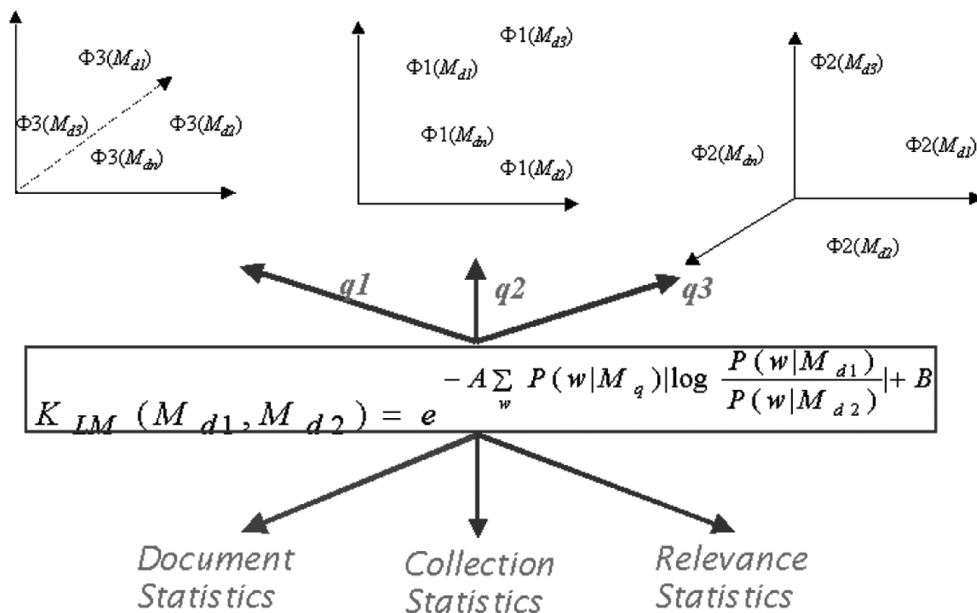


FIG. 3. The language-modeling kernel dynamically determining vector spaces.

Comparisons between the language-modeling kernel and other text kernels. Now, let us make a comparison between the language-modeling kernel and certain existing text kernels, including the Basic Vector Space Model (BVSM), the GVSM, and the LSI kernels. The comparison is made in terms of the ability to utilize term-frequency information, the ability to utilize the term-term relationship, the ability to utilize relevance statistics, and whether the kernel is an integrated kernel. A kernel is called an integrated kernel if the mechanisms to utilize the aforementioned data statistics, such as term frequency information, are directly incorporated into the kernel itself. The results of the comparison are summarized shown in Table 1.

The BVSM kernel uses the normalized inner product to evaluate the similarity relationship between two documents. The kernel itself does not tell us how to use term-frequency information. In other words, before applying this kernel, one first has to utilize some other heuristics such as TF-IDF or BM25 to index the terms for each document. In this sense, we do not count it as integrated kernel. The BVSM kernel can be expressed as:

$$K_{BVSM} = \frac{\langle D1, D2 \rangle}{\sqrt{\langle D1, D1 \rangle \langle D2, D2 \rangle}}$$

The GVSM kernel (Carbonell et al., 1997; Wong et al., 1985; Wong, Ziarko, Raghavan, & Wong, 1989) takes the term-term relationship into consideration. Intuitively, if two terms co-occur frequently in the same documents, their similarity relationship is strong. The GVSM kernel can be expressed as:

$$K_{GVSM} = \frac{\langle A^T D1, A^T D2 \rangle}{\sqrt{\langle A^T D1, A^T D1 \rangle \langle A^T D2, A^T D2 \rangle}}$$

where A is term-document matrix. For the same reason as that for the BVSM, the GVSM is not an integrated kernel.

The LSI kernel (Cristianini, Shawe-Taylor, & Lodhi, 2002; Deerwester et al., 1990) maps documents to a vector space, by utilizing latent semantic structures as orthogonal dimensions. Those latent semantic structures are linear combinations of document vectors, which are computed by applying the SVD on the term-document matrix A . Formally, the LSI kernel can be expressed as:

$$A = U \Sigma V^T;$$

$$K_{LSI} = \frac{\langle U^T D1, U^T D2 \rangle}{\sqrt{\langle U^T D1, U^T D1 \rangle \langle U^T D2, U^T D2 \rangle}},$$

where U is the matrix of eigenvectors derived from AA^T , Σ is an diagonal matrix of singular values stored in the descending order, and V is the matrix of eigenvectors derived from $A^T A$. In fact, the LSI kernel also utilizes term-term co-occurrence information embedded in the document collection. However, just like the BVSM and the GVSM kernels, the LSI kernel also fails to model the user's information need (i.e., relevance statistics). Moreover, it cannot be viewed as an integrated kernel because this kernel itself does not describe how to use term-frequency information.

As can be seen, only the proposed language-modeling kernel has the ability to integrate all three types of information, term frequency, term-term occurrence, and relevance statistics, into a unified framework. Therefore, the vector space determined by the language-modeling kernel also is able to incorporate all these three types of information. In other words, the language-modeling kernel provides a systematic way to build an optimal vector space for information retrieval. By Mercer's theory, the eigenfunctions of the language-modeling kernel act as the features of the mapped vector space.

Retrieval Process Based on the Language-Modeling Kernel

As shown in Figure 1, the interface of the language-modeling kernel based approach is the vector space determined by the language-modeling kernel. According to Mercer's theory, the inner product in the mapped vector space is equal to the language-modeling kernel in the original space. Therefore, the ranking function of our approach at the initial retrieval stage will be:

$$K_{LM}(M_Q, M_{Di}) = e^{-A \sum_w P(w|M_Q) \log \frac{P(w|M_{Di})}{P(w)}} + B. \quad (3)$$

After the initial result is generated, at the learning stage, a unique double learning strategy is proposed. First, positive samples (i.e., relevant documents specified by the user) are used to reestimate query language model $P(w|M_Q)$. Then, unlike the model-based feedback approach (Zhai & Lafferty, 2001a), which directly applies the reestimated query model to the initial ranking function, our approach builds the language-modeling kernel K_{LM} (shown in Equation 2) based on the learned query language model. Finally, the SVM is applied in the mapped vector space that is determined by the kernel K_{LM} to find an optimal decision boundary that separates the relevant documents from the nonrelevant ones. As one can see, unlike the model-based feedback approach, which is only able to use positive samples, the language-modeling kernel based approach can utilize both positive and negative samples. The decision boundary that the SVM generates in the mapped vector space based on the training data can be expressed as follows:

TABLE 1. Comparison between the language-modeling kernel and existing text kernels.

Text kernel	Ability to utilize term frequency	Ability to utilize term-term relationship	Ability to utilize relevance statistics	Integrated kernel
BVSM kernel	+	-	-	-
GVSM kernel	+	+	-	-
LSI kernel	+	+	-	-
LM kernel	+	+	+	+

$$f(M_D) = \sum_i a_i y_i K_{LM}(M_{D_i}, M_D) + b, \quad (4)$$

where D_i is one of the feedback documents. If D_i is relevant, $y_i = 1$; otherwise $y_i = -1$. By this decision boundary, a new Document D will be judged as relevant if $f(M_D) > 0$; and nonrelevant if $f(M_D) \leq 0$. Since the kernel component of this decision boundary is equal to the inner product in the mapped vector space, even if we do not know the mapping function Φ , we can still run machine learning algorithms on the mapped vector space obtained via the kernel function. This is known as the *kernel trick*.

Initial retrieval stage. There are different ways to estimate the query language model $P(w|M_Q)$ and the document language model $P(w|M_{D_i})$ for the initial ranking function 3. The simplest one is to directly estimate them from Query Q and Document D_i using the MLE. Since the query string always is much shorter than documents, we have $\hat{P}(w|M_Q) \gg \hat{P}(w|M_{D_i})$ for $w \in Q$. Therefore, the ranking function reduces to:

$$K_{LM}(M_Q, M_{D_i}) \propto \sum_{w \in Q} P(w|M_{D_i}), \quad (5)$$

which is the ranking function used by the original language modeling approach proposed in Ponte and Croft (1998).

A more sophisticated way is to take term-term co-occurrence information into consideration when estimating $P(w|M_Q)$ for the ranking function 3. For example, we can use the Markov chain expansion to estimate the query language model (Lafferty & Zhai, 2001). For this approach, if we only keep the words that satisfy $P(w|M_Q) > threshold$, and assume $P(w|M_Q) > P(w|M_{D_i})$ holds for those words, then the ranking function 3 reduces to:

$$K_{LM}(M_Q, M_{D_i}) \propto \sum_w [P(w|M_Q) \log P(w|M_{D_i})], \quad (6)$$

which is the ranking function used in (Lafferty & Zhai, 2001).

Learning stage. The proposed language-modeling kernel based approach utilizes a double learning strategy to learn from the user's relevance feedback. For the first learning, the query language model is refined by using the positive samples, and the language-modeling kernel is dynamically built based on the refined query language model. The second learning applies the language-modeling kernel based SVM to find an optimal decision boundary that separates the positive samples from the negative samples. Finally, the learned decision boundary, combined with documents' initial Retrieval Status Values (i.e., numbers indicating how well a document matches the query), is used to re-rank all the documents in the document collection.

Since relevant documents match the user's information need, it is natural to refine the query language model based on the relevant documents at the first learning stage. Two strategies for this purpose were proposed in Zhai and Lafferty (2001a). One is called divergence minimization (div-min); that is, the query model is estimated by minimizing the average divergence over the document language models of the relevant documents. The other is called the generative mixture model (i.e., mixture), where the expectation-maximization algorithm (McLachlan & Krishnan, 1997) is used to estimate the query language model from which the relevant documents are generated. These two approaches represent state-of-the-art language modeling techniques that utilize relevance feedback; however, neither of these methods can take advantage of negative feedback information.

Unlike the model-based feedback approach that puts the refined query language model back to the original ranking function, our approach utilizes the refined query language model to dynamically build the language-modeling kernel shown in Equation 2, and applies the SVM, in the mapped vector space determined by the kernel, to find a decision boundary that separates the relevant documents from the nonrelevant ones. Therefore, our approach is able to not only take advantage of both positive and negative samples but also to integrate the SVM into language modeling techniques. In Figure 4, we present the whole picture of the language modeling kernel based approach for information retrieval.

Results

As shown earlier, the language-modeling kernel is a unified model that smoothly incorporates the advantages of the support vector machine with the language modeling techniques. In this section, we will demonstrate that this type of incorporation significantly outperforms the learning abilities that the language modeling techniques currently possess.

We use different large TREC plain text and Web collections as test datasets, including the TREC7 ad hoc task collection (Disks 4 & 5 without CR, Topics 351–400), TREC8 ad hoc task collection (Disks 4 & 5 without CR, Topics 401–450), TREC9 WEB main task collection (WT10G, 1.69 million Web documents, Topics 451–500), and TREC2001 WEB topic relevant task collection (WT10G, Topics 501–550).

The focus of our experiments is to compare the learning performance provided by the proposed language-modeling kernel to the best performance obtainable using the learning capabilities that the language modeling techniques currently possess. To test learning, certain feedback mechanisms should be used. A frequently used mechanism is called *pseudo-relevance feedback*, which simply assumes that the top n ranking documents returned by the initial retrieval are relevant. Nevertheless, the problem of using pseudo-relevance feedback to compare different learning strategies is that the learning result based on pseudo-relevance feedback largely depends on the result quality of the initial retrieval step, instead of just the learning strategy itself. If a certain amount of the top n ranking documents returned by the initial

Algorithm 1 Language-Modeling Kernel Based Approach

- 1: **for** each document D_i in Collection C **do**
- 2: $RSV_i = K_{LM}(M_Q, M_{D_i})$, where

$$K_{LM}(M_Q, M_{D_i}) = e^{-A \sum_w P(w|M_Q) |\log \frac{P(w|M_Q)}{P(w|M_{D_i})}| + B}.$$

{This is formula 3, which can be estimated by using either formula 5 or 6.}

- 3: **end for**
- 4: Sort collection C in the descending order of RSV .
- 5: Estimate query language model $P(w|M_Q)$ from relevant documents by using either generative mixture model (mixture) or divergence minimization approach (div-min).
- 6: Calculate the kernel function shown in formula 2:

$$K_{LM}(M_{D_1}, M_{D_2}) = e^{-A \sum_w P(w|M_Q) |\log \frac{P(w|M_{D_1})}{P(w|M_{D_2})}| + B}.$$

- 7: Utilize relevant documents as positive training data and non-relevant documents as negative training data, and apply the SVM on the training data to find the optimal decision boundary.
 - 8: **for** each document D_i in Collection C **do**
 - 9: $f(M_{D_i}) = \sum_j a_j y_j K_{LM}(M_{D_j}, M_{D_i}) + b$. {formula 4}
 - 10: $RSV_i = RSV_i + f(M_{D_i})$.
 - 11: **end for**
 - 12: Re-rank collection C in the descending order of RSV .
-

FIG. 4. Algorithm 1: Language-modeling kernel based approach.

retrieval are actually not relevant, the comparison of learning strategies based on the assumption that these n documents are relevant will become meaningless. Therefore, to isolate the impact of the learning performance, we adopt the option of using real feedback. By showing that we achieve superior performance in this context, we can conclude that the proposed mechanism would benefit the retrieval process regardless of the way in which feedback is provided (i.e., pseudo, real, or implicit), as long as the quality of the feedback given to the system is reasonable. Since the set of relevant documents is actually available for each TREC query, we select the top n (n is a small number.) relevant documents from the initial retrieval result list as the positive feedback and those irrelevant documents that rank higher than the n th relevant document are deemed as the negative feedback.

In detail, for each test collection, the titles of the topic descriptions are used as queries. At the initial retrieval stage, ranking function 5 (i.e., the basic ranking function used by the original language modeling approach) was applied to obtain 2,000 initial results. At the learning stage, the top n (In our experiment, n will be 5, 10, respectively.) ranking relevant documents are specified as positive samples; and up to 20 unspecified documents that rank higher than the n th specified document, if there are any, are used as negative samples. Two learning strategies are applied to re-rank the top 2,000 initial results. One is our proposed language-modeling kernel based learning strategy as described earlier; the other one used for comparison is the state-of-the-art, model-based feedback mechanism (Zhai & Lafferty, 2001a). We used the Lemur IR toolkit (<http://www.lemurproject.org>) to generate

the results for the comparison strategy. Both learning strategies need to refine the query language model $P(w|M_Q)$ from the user's positive feedback. For this purpose, two estimating techniques are used: One is the generative mixture model (i.e., mixture), and the other is the divergence minimization (div-min). In summary, on each TREC collection, we run all possible combinations from the following options:

- Parameter n : 5 versus 10.
- Learning strategy: the language-modeling kernel based approach versus the model-based feedback approach.
- Query model refining technique: mixture versus div-min.

At each run, three standard measures—*average precision*, *initial precision* (i.e., precision at recall 0% in the interpolated precision-recall curve), and the *interpolated precision-recall curve*—among the top 1,000 results were evaluated. Statistical analysis based on the experimental results show that the language-modeling kernel based approach outperforms the model-based feedback approach on all these collections. For the WT10G WEB collection, the improvements made by the language-modeling kernel based approach are significant.

Results on TREC WEB Collection WT10G

This subsection shows the performance of both the language-modeling kernel based approach and the model-based feedback approach on a large WEB collection, TREC WT10G.

TABLE 2. Comparison of the language-modeling kernel based approach (LM kernel) and the model-based feedback approach (Model based) on TREC WT10G.

Collection	<i>n</i>	Precision type	div-min			mixture			
			Model based	LM kernel	%Improvement (1)	Model based	LM kernel	%Improvement (2)	%Improvement (3)
TREC 2001	10	AvePr.	0.3060	0.3939	+28.7	0.3685	0.4691	+27.3	+27.3
		InitPr.	0.8552	0.9717	+13.6	0.9693	1	+3.2	+3.2
	5	AvePr.	0.3069	0.3257	+6.1	0.2997	0.4098	+36.7	+33.5
		InitPr.	0.8348	0.9655	+15.7	0.9063	1	+10.3	+10.3
TREC 09	10	AvePr.	0.3097	0.3694	+19.4	0.3644	0.4925	+35.2	+35.2
		InitPr.	0.8266	0.9248	+11.9	0.8994	0.995	+10.6	+10.6
	5	AvePr.	0.3206	0.2844	-11.3	0.3612	0.4529	+25.4	+25.4
		InitPr.	0.8405	0.8103	-3.6	0.9286	0.996	+7.3	+7.3

TREC Web collection WT10G was used in the TREC-9 and TREC 2001 Web Tracks (Voorhees, 2001; Voorhees & Harman, 2000). WT10G was measured to be like the actual Web in terms of power law relationships, diameter, and connected components (Soboroff, 2002). Therefore, the experiments on WT10G can be used to evaluate the potential of our language-modeling kernel based approach for Web searching. Table 2 records the experimental results in terms of average precision (AvePr.) and initial precision (InitPr.) over 50 queries on both the TREC09 WEB main task collection and the TREC2001 WEB topic relevant task collection. We notice the following facts from the experimental results:

- If the generative mixture model (mixture) is applied to refine the query language model, both the language-modeling kernel based approach and the model-based feedback approach perform better than if the divergence minimization (div-min) model is applied.
- If the same query model refining technique (either div-min or mixture) is used, the language-modeling kernel based approach outperforms the model-based feedback approach, except for one case where div-min is applied to refine the query language model based on only five positive samples on the TREC9 WEB collection. The improvements made by the language-modeling approach over the model-based feedback approach are shown in column “Improvement (1)” in Table 2 when div-min is used as the query model refining technique and are shown in column “Improvement (2)” in Table 2 when mixture is used as the query model refining technique.
- No matter which query model refining technique is used, even the worst results of the language-modeling kernel based approach outperform the best results of the model based feedback approach, except for one case where only five positive samples are provided on the TREC9 WEB collection.
- The best results of the language-modeling kernel based approach significantly outperform the best results of the model-based feedback approach. The improvements made by the best results of the language-modeling kernel based approach over the best results of the model-based feedback approach are shown in column “Improvement (3)” in Table 2.

Besides average precision and initial precision over the 50 queries for each collection, the language-modeling kernel based approach also makes significant improvements on

precision over the 50 queries at each recall level. The PR curves for the TREC2001 WEB collection are shown in Figure 5 (with 10 positive feedbacks) and Figure 6 (with five positive feedbacks) while the PR curves for the TREC09 WEB collection are shown in Figure 7 (with 10 positive feedbacks) and Figure 8 (with five positive feedbacks). In each figure, the two solid curves are generated by the language-modeling kernel based approach.

Results on the Large TREC Plain Text Collections

This subsection shows the performance of both the language-modeling kernel based approach and the model-based feedback approach on the large TREC plain text collections.

TREC plain text documents are distributed on five CDs with approximately 1 GB on each. Both TREC7 and TREC8 use Disk 4–5, excluding “Congressional Record” as the test set (Voorhees & Harman, 1998, 1999).

Table 3 records the experimental results in terms of average precision (AvePr.) and initial precision (InitPr.) over 50 queries on both the TREC7 ad hoc task collection and the

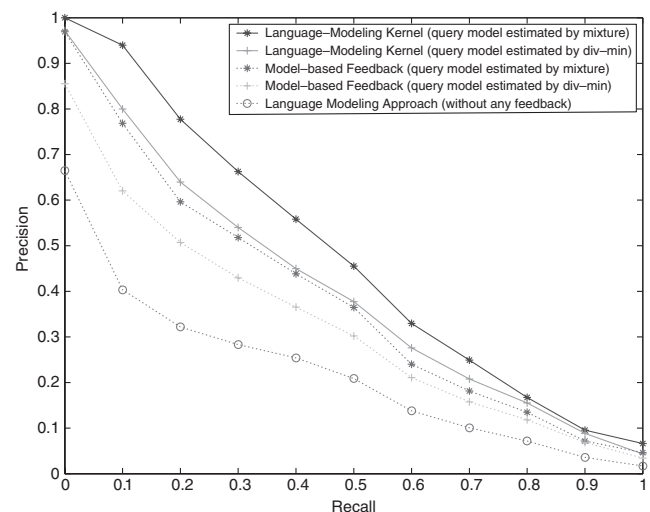


FIG. 5. The precision-recall curve on the TREC2001 WEB collection with 10 positive feedbacks.

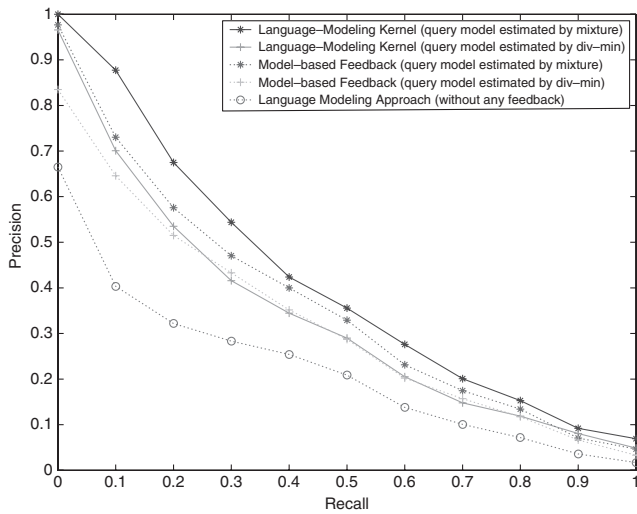


FIG. 6. The precision-recall curve on the TREC2001 WEB collection with 5 positive feedbacks.

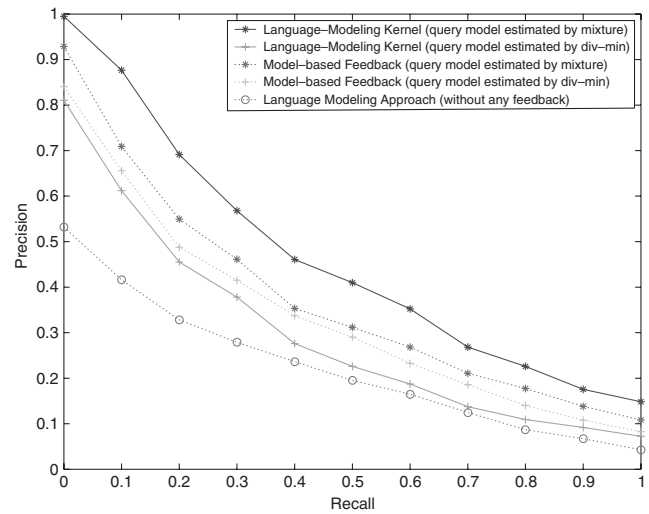


FIG. 8. The precision-recall curve on the TREC9 WEB collection with 5 positive feedbacks.

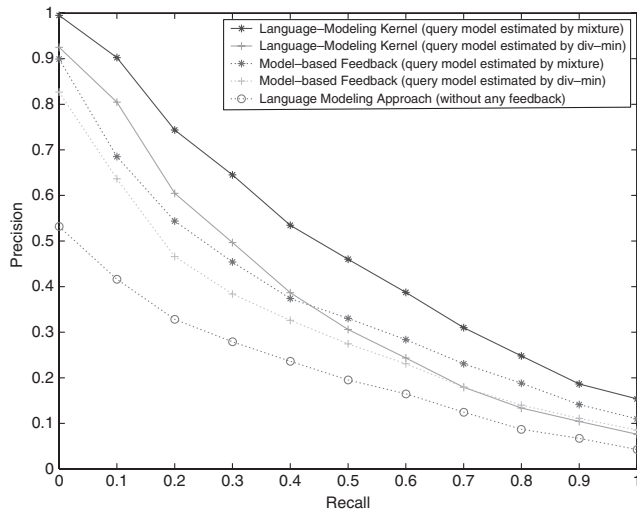


FIG. 7. The precision-recall curve on the TREC9 WEB collection with 10 positive feedbacks.

TREC8 ad hoc task collection. We notice the following facts from the experimental results:

- If the generative mixture model (mixture) is applied to refine the query language model, both the language-modeling kernel based approach and the model-based feedback approach perform better than if the divergence minimization (div-min) model is applied.
- If the same query model refining technique (either div-min or mixture) is used, the language-modeling kernel based approach outperforms the model-based feedback approach. The improvements made by the language-modeling approach over the model-based feedback approach are shown in column “Improvement (1)” in Table 3 when div-min is used as the query model refining technique and are shown in column “Improvement (2)” in Table 3 when mixture is used as the query model refining technique.
- No matter which query model refining technique is used, the best results of the language-modeling kernel based approach outperform the best results of the model-based feedback approach. The improvements made by the best results of the

TABLE 3. Comparison of the language-modeling kernel based approach (LM kernel) and the model-based feedback approach (Model based) on large TREC plain text collections.

Collection	n	Precision type	div-min			mixture			%Improvement (3)
			Model based	LM kernel	%Improvement (1)	Model based	LM kernel	%Improvement (2)	
TREC 07	10	AvePr.	0.2098	0.2703	+28.8	0.3455	0.3817	+10.5	+10.5
		InitPr.	0.7771	0.9471	+21.9	0.9933	1	+0.7	+0.7
	5	AvePr.	0.2131	0.2406	+12.9	0.3192	0.3354	+5.1	+5.1
		InitPr.	0.7937	0.9390	+19.8	1	1	0	0
TREC 08	10	AvePr.	0.2635	0.3391	+28.7	0.3687	0.4090	+11	+11
		InitPr.	0.7620	0.9581	+25.7	1	1	0	0
	5	AvePr.	0.2699	0.3045	+12.8	0.3373	0.3514	+4.2	+4.2
		InitPr.	0.7839	0.9518	21.4	1	1	0	0

language-modeling kernel based approach over the best results of the model-based feedback approach are shown in column “Improvement (3)” in Table 3.

Besides average precision and initial precision over the 50 queries for each collection, the language-modeling kernel based approach also makes significant improvements on precision over the 50 queries at each recall level. The PR curves for the TREC7 ad hoc task collection are shown in Figure 9 (with 10 positive feedbacks) and Figure 10 (with five positive feedbacks) while the PR curves for the TREC8 ad hoc task collection are shown in Figure 11 (with 10 positive feedbacks) and Figure 12 (with five positive feedbacks). In each figure, the two solid curves are generated by the language-modeling kernel based approach.

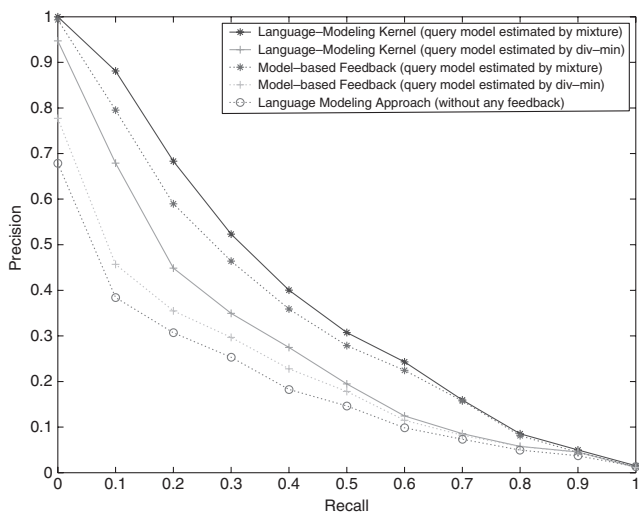


FIG. 9. The precision-recall curve on the TREC7 ad hoc task collection with 10 positive feedbacks.

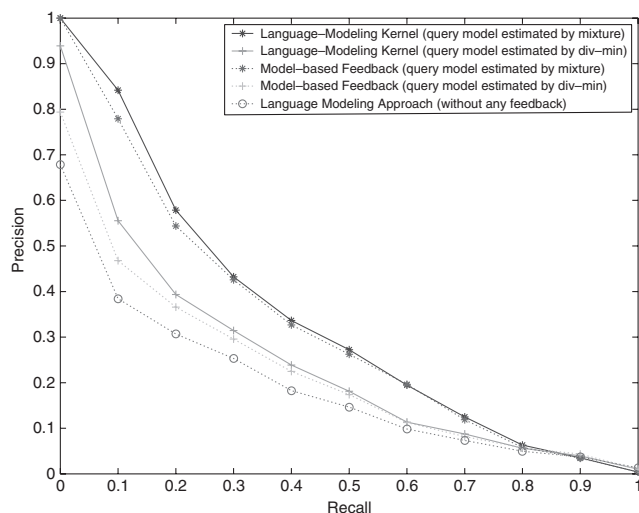


FIG. 10. The precision-recall curve on the TREC7 ad hoc task collection with 5 positive feedbacks.

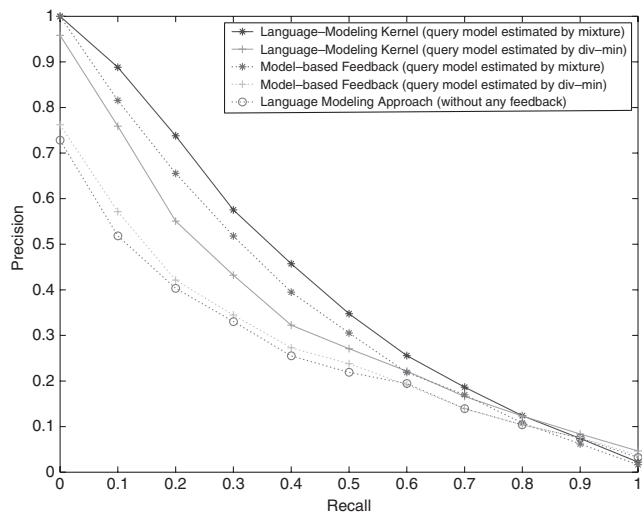


FIG. 11. The precision-recall curve on the TREC8 ad hoc task collection with 10 positive feedbacks.

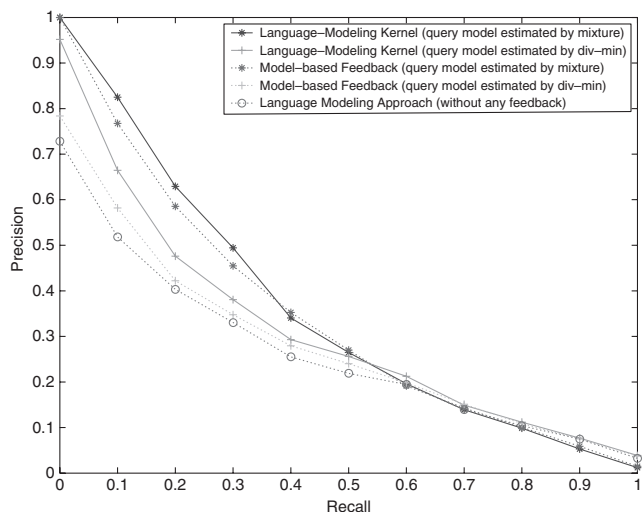


FIG. 12. The precision-recall curve on the TREC8 ad hoc task collection with 5 positive feedbacks.

Statistical Analysis of the Experimental Results

In this subsection, we compare the performance of the proposed language-modeling kernel based approach and the model-based feedback approach by using statistical hypothesis testing. Our method for hypothesis testing can be described as follows:

Let our goal be verifying if population P_1 's mean μ_1 is the same as population P_2 's mean μ_2 (i.e., verifying if $\mu_1 = \mu_2$). Therefore, we form the null hypothesis (H_0) and alternative hypothesis (H_1) as follows:

$$H_0: \mu_1 - \mu_2 = 0;$$

$$H_1: \mu_1 > \mu_2 > 0;$$

Assume that sample S_1 with size sz_1 is obtained from P_1 , and sample S_2 with size sz_2 is obtained from P_2 . Let sample S_1 have mean m_1 and standard deviation sd_1 ; sample S_2 have

TABLE 4. Statistical analysis on average precision.

		Web collection		Plain text collection	
		TREC2001	TREC09	TREC07	TREC08
LM kernel	sz_1	200	200	200	200
	m_1	0.399629086	0.399803073	0.306994566	0.350976507
	sd_1	0.22664523	0.270662595	0.206759587	0.224449502
Model based	sz_2	200	200	200	200
	m_2	0.333846669	0.338954019	0.272726604	0.309842505
	sd_2	0.202203761	0.24785895	0.243374036	0.210020188
Calculated T		3.062888801	2.344754862	1.679266994	1.892483478
Tabulated t		1.649	1.649	1.649	1.649

TABLE 5. Statistical analysis on initial precision.

		Web collection		Plain text collection	
		TREC2001	TREC09	TREC07	TREC08
LM kernel	sz_1	200	200	200	200
	m_1	0.984285714	0.931286501	0.971506573	0.977472222
	sd_1	0.078578283	0.226386926	0.149087805	0.117861864
Model based	sz_2	200	200	200	200
	m_2	0.909024501	0.873766605	0.895177453	0.886459396
	sd_2	0.204172375	0.255813718	0.243374036	0.24011181
Calculated T		4.865144655	2.381294502	3.782144597	4.812022497
Tabulated t		1.649	1.649	1.649	1.649

mean m_2 and standard deviation sd_2 . Then we compute a t value T as follows:

$$T = \frac{m_1 - m_2}{\sqrt{\frac{sz_1 + sz_2}{sz_1 sz_2} \frac{(sz_1 - 1)sd_1^2 + (sz_2 - 1)sd_2^2}{sz_1 + sz_2 - 2}}}$$

Since our alternate hypothesis (H_1) is one sided, a single-tailed t statistic is used in the testing. We use the typical level of significance $\alpha = 0.05$ with $df = sz_1 + sz_2 - 1$ to obtain the tabulated t value denoted as t from a standard t table. If $T > t$, we can reject the null hypothesis H_0 in favor of the alternate hypothesis H_1 , thereby failing to disprove the alternate hypothesis.

We use the aforementioned statistical test method to analyze the experimental results we obtained from TREC7 and TREC8 ad hoc collections, and TREC9 and TREC2001 WEB collections. For each collection, let all average precisions (or initial precisions) generated by the proposed language-modeling kernel based approach be $P1$, and all average precisions (or initial precisions) generated by the model-based feedback approach be $P2$. For each collection, we obtained the sample $S1$ from $P1$, and sample $S2$ from $P2$ by conducting a series of experiments described earlier. Both the size of sample $S1$ (denoted as sz_1) and the size of sample $S2$ (denoted as sz_2) can be calculated as: $sz_1 = sz_2 = 50$ standard queries $\times 2$ different values of parameter

n (5 vs. 10) $\times 2$ types of query model refining technique (mix vs. div-min) = 200.

The statistical analysis results for average precision are shown in Table 4, and the results for initial precision are shown in Table 5.

As can be seen from Table 4, for each collection, the calculated T is greater than the tabulated t (i.e., $T > t$). Therefore, we reject the null hypothesis in favor of the alternative hypothesis for each collection. In other words, we can say that on the average, the proposed language-modeling kernel based approach generates better average precision than does the model-based feedback approach for all the aforementioned TREC collections.

As can be seen from Table 5, for each collection, the calculated T is greater than the tabulated t (i.e., $T > t$). Therefore, we reject the null hypothesis in favor of the alternative hypothesis for each collection. In other words, we can say that on the average, the proposed language-modeling kernel based approach generates better initial precision than does the model-based feedback approach for all the aforementioned TREC collections.

Discussion

As one can see, the language-modeling kernel based approach achieves consistent improvements over the state-of-the-art model-based feedback approach on both large TREC Web collections and large TREC plain text collections.

Those consistent improvements not only experimentally justified the fact that our proposed systematic approach used to build the vector space is better than the existing strategies, such as the BVSM, the GVSM, and the LSI, but also proved the effectiveness of incorporating the SVM into the language modeling technique. Note that on the Web collections, the improvements made by the language-modeling kernel based approach are significant. These experimental results suggest that the potential for applying the language-modeling kernel based approach to Internet searching is great.

Conclusions and Future Work

We have presented an integrated information retrieval framework, where language-modeling techniques and the VSM are viewed as integral components at different levels. Language modeling techniques contribute their estimation capabilities while the VSM provides an interface where efficient learning machines such as the SVM can be applied to find an optimal boundary between relevant and nonrelevant documents. In other words, these two approaches mutually strengthen each other in this integrated framework. The key component that integrates together the language modeling technique and the VSM is the language-modeling kernel. We designed this kernel in such a way that it not only has all the modeling powers provided by the language modeling technique but also dynamically determines an optimal vector space that is tailored to the user's information need. In addition to having the ability to utilize the term-distribution information and the term-term co-occurrence information, which is crucial information for high-quality retrieval, the language-modeling kernel based approach adopts a double learning strategy. For the first learning stage, positive samples are used to refine the query language model, on which the language-modeling kernel is dynamically built, and learning at the second stage generates an optimal decision boundary between the positive and negative samples in the mapped vector space determined by the language-modeling kernel. Large-scale experiments on standard TREC text collections show the significant improvements our approach made over the state-of-the-art information retrieval method. Our future work will focus on applying the proposed language-modeling kernel to achieve personalized information retrieval.

References

- Carbonell, J.G., Yang, Y., Frederking, R.E., Brown, R.D., Geng, Y., & Lee, D. (1997). Translingual information retrieval: A comparative evaluation. *Proceedings of the 1997 International Joint Conference on Artificial Intelligence* (pp. 708–715). Nagoya: Morgan Kaufmann.
- Cristianini, N., Shawe-Taylor, J., & Lodhi, H. (2002). Latent semantic kernels. *Journal of Intelligent Information Systems*, 18, 127–152.
- Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W., & Harshman, R.A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, 391–407.
- Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. *Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 111–119). New Orleans: ACM.
- McLachlan, G.J., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: Wiley.
- Ponte, J., & Croft, W.B. (1998). A language-modeling approach to information retrieval. *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 275–281). New York: ACM.
- Soboroff, I. (2002). Does wt10g look like the web? *Proceedings of the 25th ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 423–424). Tampere: ACM.
- Voorhees, E. (2001). Overview of TREC 2001. *NIST Special Publication 500–250: The 10th Text REtrieval Conference* (pp. 1–15). Gaithersbury: NIST.
- Voorhees, E., & Harman, D. (1998). Overview of the 7th Text REtrieval Conference (TREC-7). *NIST Special Publication No. 500–242: The 7th Text REtrieval Conference* (pp. 1–24). Gaithersbury: NIST.
- Voorhees, E., & Harman, D. (1999). Overview of the 8th Text REtrieval Conference (TREC-8). *NIST Special Publication No. 500–246: The 8th Text REtrieval Conference* (pp. 1–24). Gaithersbury: NIST.
- Voorhees, E., & Harman, D. (2000). Overview of the 9th Text REtrieval Conference (TREC-9). *NIST Special Publication No. 500–249: The 9th Text REtrieval Conference* (pp. 1–14). Gaithersbury: NIST.
- Wong, S.K.M., Ziarko, W., Raghavan, V.V., & Wong, P.C.N. (1989). Extended Boolean query processing in the generalized vector space model. *Information Systems*, 14, 47–63.
- Wong, S.K.M., Ziarko, W., & Wong, P.C.N. (1985). Generalized vector spaces model in information retrieval. *Proceedings of the 8th annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 18–25). Montreal: ACM.
- Zhai, C., & Lafferty, J. (2001a). Model-based feedback in the language-modeling approach to information retrieval. *Proceedings of the 10th International Conference on Information and Knowledge Management* (pp. 403–410). Atlanta: ACM.
- Zhai, C., & Lafferty, J. (2001b). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 2, 334–342.