Document Language Models, Query Models, and Risk Minimization for Information Retrieval

John Lafferty
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Chengxiang Zhai
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

We present a framework for information retrieval that combines document models and query models using a probabilistic ranking function based on Bayesian decision theory. The framework suggests an operational retrieval model that extends recent developments in the language modeling approach to information retrieval. A language model for each document is estimated, as well as a language model for each query, and the retrieval problem is cast in terms of risk minimization. The query language model can be exploited to model user preferences, the context of a query, synonomy and word senses. While recent work has incorporated word translation models for this purpose, we introduce a new method using Markov chains defined on a set of documents to estimate the query models. The Markov chain method has connections to algorithms from link analysis and social networks. The new approach is evaluated on TREC collections and compared to the basic language modeling approach and vector space models together with query expansion using Rocchio. Significant improvements are obtained over standard query expansion methods for strong baseline TF-IDF systems, with the greatest improvements attained for short queries on Web data.

1. INTRODUCTION

The language modeling approach to information retrieval has recently been proposed as a new alternative to traditional vector space models and other probabilistic models. In the use of language modeling by Ponte and Croft [17], a unigram language model is estimated for each document, and the likelihood of the query according to this model is used to score the document for ranking. Miller *et al.* [15] smooth the document language model with a background model using hidden Markov model techniques, and demonstrate good performance on TREC benchmarks. Berger and Lafferty [1] use methods from statistical machine translation to incorporate synonomy into the document language model, achieving effects similar to query expansion in more standard approaches to IR. The relative simplicity and effectiveness of the language modeling approach, together with the fact that it leverages statistical methods that have been developed in speech recognition and other areas, makes it an attractive framework in which to develop new text retrieval methodology.

In this paper we motivate the language modeling approach from a general probabilistic retrieval framework based on risk minimization. This framework not only covers the classical probabilistic retrieval models as special cases, but also suggests an extension of the existing language modeling approach to retrieval that involves estimating both document language models and query language models and comparing the models using the Kullback-Leibler divergence. In the case where the query language model is concentrated on the actual query terms, this reduces to the ranking method employed by Ponte and Croft [17] and others. We also introduce a novel method for estimating an expanded query language model, which may assign probability to words that are not in the original query. The essence of the new method is a Markov chain word translation model that can be computed based on a set of documents. The Markov chain method is a very general method for expanding either a query model or a document model. As a translation model, it addresses several basic shortcomings of the translation models used by Berger and Lafferty [1], as described in Section 4. The query models explored in this paper are quite simple, but in general, the role of the query model is to incorporate knowledge of the user and the context of an information need into the retrieval model.

The paper is organized as follows. In Section 2 we discuss the language modeling approach to IR, and briefly review previous work in this direction. In Section 3 we present the risk minimization retrieval framework and our extension to the language modeling approach that incorporates both query and document language models. Section 4 presents the idea of using Markov chains on a documents and words to expand document and query models, and gives several examples. This technique requires various collection statistics to be calculated, and we explain in Section 5 how these can be calculated at index time. A series of experiments to evaluate these methods is presented in Section 6, where we attempt to compare directly to state-of-the art ranking functions and weighting schemes. Conclusions and the contributions of this work are summarized in Section 8.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA
Copyright 2001 ACM 1-58113-331-6/01/0009 ...$5.00.

2. THE LANGUAGE MODELING APPROACH

In the language modeling approach to information retrieval, a multinomial model $p(w \mid \mathbf{d})$ over terms is estimated for each document $\mathbf{d}$ in the collection $\mathcal{C}$ to be indexed and searched. This model is used to assign a likelihood to a user's query $\mathbf{q} = (q_1, q_2, \ldots, q_m)$. In the simplest case, each query term is assumed to be independent of the other query terms, so that the query likelihood is given by $p(\mathbf{q} \mid \mathbf{d}) = \prod_{i=1}^{m} p(q_i \mid \mathbf{d})$. After the specification of a document prior $p(\mathbf{d})$, the *a posteriori* probability of a document is given by

$$p(\mathbf{d} \mid \mathbf{q}) \propto p(\mathbf{q} \mid \mathbf{d})\, p(\mathbf{d})$$

and is used to rank the documents in the collection $\mathcal{C}$.

Just as in the use of language models for speech recognition, language models for information retrieval must be "smoothed," so that non-zero probability can be assigned to query terms that do not appear in a given document. One of the simplest ways in which a document language model can be smoothed is by linear interpolation with a background collection model $p(w \mid \mathcal{C})$:

$$p_\lambda(w \mid \mathbf{d}) = \lambda\, p(w \mid \mathbf{d}) + (1 - \lambda)\, p(w \mid \mathcal{C}) \tag{1}$$

Miller *et al.* [15] view this smoothed model as coming from a simple 2-state hidden Markov model, and train the parameter λ using maximum likelihood estimation. One of the main effects of this type of smoothing is robust estimation of common, content-free words that are typically treated as "stop words" in many IR systems.

A potentially more significant and effective kind of smoothing is what may be referred to as *semantic smoothing*, where synonyms and word sense information is incorporated into the models. With proper semantic smoothing, a document that contains the term $w =$ `automobile` may be retrieved to answer a query that includes the term $q =$ `car`, even if this query term is not present in the document. Semantic smoothing effects are achieved in more standard approaches to IR using query expansion and relevance and pseudo-relevance feedback techniques. The development of a well-motivated framework for semantic smoothing is one of the important unresolved problems in the language modeling approach.

In order to incorporate a kind of semantic smoothing into the language modeling approach, Berger and Lafferty [1] estimate translation models $t(q \mid w)$ for mapping a document term w to a query term q. Using translation models, the document-to-query model becomes

$$p(\mathbf{q} \mid \mathbf{d}) = \prod_{i=1}^{m} \sum_{w} t(q_i \mid w)\, p(w \mid \mathbf{d})$$

Berger and Lafferty [1] report significant improvements over the baseline language modeling approach through the use of translation models.

One of the primary motivations of the present paper is to address what we view as several difficulties with the translation model approach to semantic smoothing in the language modeling framework. First, the translation models $t(q \mid w)$ must be estimated from training data. As the models are highly lexical, it is unlikely that a sufficiently large collection of relevance judgments will be available to estimate them on actual user data. Because of this, Berger and Lafferty generate an artificial collection of "synthetic" data for training. Second, the application of translation models to ranking is inefficient, as the model involves a sum over all terms in the document. Third, the translation probabilities are context-independent, and are therefore unable to directly incorporate word-sense information and context into the language models.

In the following section we present a formal retrieval framework based on risk minimization and derive an extension of the language modeling approach just described, which may ultimately be better suited to semantic smoothing to model the user's information need. In Section 4 we then present a technique for expanding document and query models that addresses some of the shortcomings of the translation models as used in [1].

3. A RISK MINIMIZATION RETRIEVAL FRAMEWORK

In an interactive retrieval system, the basic action of the system can be regarded as presenting a document or a sequence of documents to the user. Intuitively, the choice of which documents to present should be based on some notion of *utility*. In this section we formalize this intuition by presenting a framework for the retrieval process based on Bayesian decision theory.

We view a query as being the output of some probabilistic process associated with the user $\mathcal{U}$, and similarly, we view a document as being the output of some probabilistic process associated with an author or document source $\mathcal{S}$. A query (document) is the result of choosing a model, and then generating the query (document) using that model. A set of documents is the result of generating each document independently, possibly from a different model. (The independence assumption is not essential, and is made here only to simplify the presentation.) The query model could, in principle, encode detailed knowledge about a user's information need and the context in which they make their query. Similarly, the document model could encode complex information about a document and its source or author.

More formally, let θ_Q denote the parameters of a query model, and let θ_D denote the parameters of a document model. A user $\mathcal{U}$ generates a query by first selecting θ_Q, according to a distribution $p(\theta_Q \mid \mathcal{U})$. Using this model, a query $\mathbf{q}$ is then generated with probability $p(\mathbf{q} \mid \theta_Q)$. Similarly, the source selects a document model θ_D according to a distribution $p(\theta_D \mid \mathcal{S})$, and then uses this model to generate a document $\mathbf{d}$ according to $p(\mathbf{d} \mid \theta_D)$. Thus, we have Markov chains $\mathcal{U} \rightarrow \theta_Q \rightarrow \mathbf{q}$ and $\mathcal{S} \rightarrow \theta_D \rightarrow \mathbf{d}$.

If $\mathcal{C} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_k\}$ is a collection of documents obtained from source $\mathcal{S}$, we denote by θ_i the model that generates document $\mathbf{d}_i$. Assume now that for each document $\mathbf{d}_i$, there is a hidden binary relevance variable R_i that depends on θ_Q and θ_i according to $p(R_i \mid \theta_Q, \theta_i)$, which is interpreted as representing the true relevance status of $\mathbf{d}_i$ with respect to $\mathbf{q}$ (1 for relevant and 0 for non-relevant). The random variable R_i is observed when we have the user's relevance judgment on $\mathbf{d}_i$, and is unobserved otherwise. In the following presentation, we will assume that R_i is not observed.

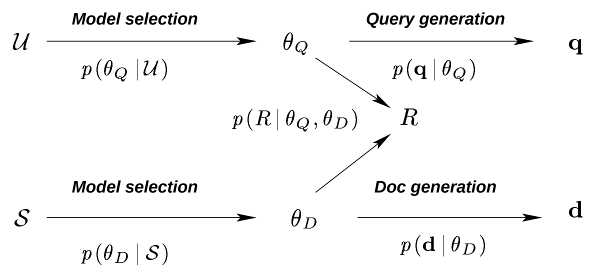

Figure 1: Generative models for queries, documents, and relevance.

Note that because the query model θ_Q can encode detailed knowledge about the user $\mathcal{U}$, the distribution of this relevance variable is in general user-specific.

A possible action of the system corresponds to a list of documents to return to the user who has issued query $\mathbf{q}$. In the general framework of Bayesian decision theory, to each such action a there is associated a *loss* $L(a, \theta)$, which in general depends upon all of the parameters of our model, $\theta \equiv (\theta_Q, \{\theta_i\}_{i=1}^k, \{R_i\}_{i=1}^k)$. In this framework, the *expected risk of action* a is given by

$$R(a \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C}) \;=\; \int_\Theta L(a, \theta)\, p(\theta \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C})\, d\theta$$

where the posterior distribution is given by

$$p(\theta \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C}) \;\propto\; p(\theta_Q \,|\, \mathbf{q}, \mathcal{U}) \prod_{i=1}^k p(\theta_i \,|\, d_i, \mathcal{S})\, p(R_i \,|\, \theta_Q, \theta_i)$$

The Bayesian decision rule is then to present the document list a^* having the least expected risk:

$$a^* \;=\; \arg\min_a R(a \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C})$$

Now, if we assume that a possible action is to return a *single* document $a = d_i$, and that the loss function only depends on θ_Q, θ_i, and R_i, the risk can be simplified to

$$R(\mathbf{d}_i; \mathbf{q}) \;\stackrel{\text{def}}{=}\; R(a = d_i \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C}) \;= \qquad (2)$$
$$\sum_{R \in \{0,1\}} \int_{\Theta_Q} \int_{\Theta_D} L(\theta_Q, \theta_D, R) \;\times$$
$$p(\theta_Q \,|\, \mathbf{q}, \mathcal{U})\, p(\theta_D \,|\, d_i, \mathcal{S})\, p(R \,|\, \theta_Q, \theta_D)\, d\theta_D\, d\theta_Q$$

Clearly, if the whole collection $\mathcal{C}$ is presented by making k sequential decisions, the result will be a list of documents ranked in ascending order of $R(\mathbf{d}_i, \mathbf{q})$. Equation 2 is our basic retrieval formula based on risk minimization.

Note that the independence assumption on the loss function does not usually hold in practice. Indeed, if we want to account for the similarity or dissimilarity between the documents in the returned list (consider, for example, the maximum marginal relevance ranking criterion proposed in [6]), such a loss function will not be appropriate. Here we make this assumption mainly for mathematical convenience.

We have deliberately left the loss function unspecified in order to achieve generality. We will be able to show that many existing operational retrieval models are special cases of the risk minimization framework when a specific loss function is used. A complete specification of the loss function will generally force us to make explicit the assumed ranking criterion and notion of relevance.

3.1 Relevance-based Loss Functions

We now consider the special case where the loss function L depends only on the relevance variable R; in this case we can simplify the risk formula and obtain a general ranking criterion based on the "probability of relevance."

Let L be defined

$$L(\theta_Q, \theta_D, R) \;=\; \begin{cases} c_0 & \text{if } R = 0 \\ c_1 & \text{if } R = 1 \end{cases}$$

where, c_0 and c_1 are two cost constants. Then we have

$$\begin{aligned} R(\mathbf{d}; \mathbf{q}) &= c_0\, p(R = 0 \,|\, \mathbf{q}, \mathbf{d}) + c_1\, p(R = 1 \,|\, \mathbf{q}, \mathbf{d}) \\ &= c_0 + (c_1 - c_0)\, p(R = 1 \,|\, \mathbf{q}, \mathbf{d}) \end{aligned}$$

This means that the risk minimization ranking criterion is now equivalent to ranking based on $p(R = 1 \,|\, \mathbf{q}, \mathbf{d})$, i.e., the probability of relevance given $\mathbf{q}$ and $\mathbf{d}$. This is the basis of all the classical probabilistic retrieval models. For example, the derivation of the binary independent model based on $p(R = 1 \,|\, \mathbf{q}, \mathbf{d})$ can be found in [18].

Interestingly, the model implicitly used in the language modeling approach can also be derived based on the probability of relevance $p(R = 1 \,|\, \mathbf{q}, \mathbf{d})$. See [14] for details of this derivation. This shows that both the classical probabilistic retrieval model and the language modeling approach to retrieval are special cases of the risk minimization framework.

3.2 Distance-based Loss Functions

We now consider another special case, where the loss function L is assumed to depend only on θ_Q and θ_D; thus, given θ_Q and θ_D, it does *not* depend on R. We will see that this allows us to derive a general probabilistic distance/similarity retrieval model.

Formally, let L be proportional to a distance or similarity measure Δ between θ_Q and θ_D, i.e.,

$$L(\theta_Q, \theta_D, R) = c\Delta(\theta_Q, \theta_D)$$

where c is a cost constant. Intuitively, if the models θ, θ' are closer/similar, then $\Delta(\theta, \theta')$ should be small. With this loss function, we have

$$R(\mathbf{d}; \mathbf{q}) \;\propto\; \int_{\theta_Q} \int_{\theta_D} \Delta(\theta_Q, \theta_D)\, p(\theta_Q \,|\, \mathbf{q}, \mathcal{U})\, p(\theta_D \,|\, \mathbf{d}, \mathcal{S}) d\theta_D\, d\theta_Q$$

This means that the risk minimization ranking criterion is now equivalent to ranking based on the expected model distance. Rather than explicitly computing this distance, we can approximate it by its value at the posterior mode:

$$\begin{aligned} R(\mathbf{d}; \mathbf{q}) &\propto \Delta(\widehat{\theta}_{\mathbf{q}}, \widehat{\theta}_{\mathbf{d}})\, p(\theta_{\mathbf{d}} \,|\, \mathbf{d}, \mathcal{S})\, p(\theta_{\mathbf{q}} \,|\, \mathbf{q}, \mathcal{U}) \\ &\propto \Delta(\widehat{\theta}_{\mathbf{q}}, \widehat{\theta}_{\mathbf{d}})\, p(\theta_{\mathbf{d}} \,|\, \mathbf{d}, \mathcal{S}) \end{aligned}$$

where

$$\begin{aligned} \widehat{\theta}_{\mathbf{q}} &= \arg\max_{\theta_Q} p(\theta_Q \,|\, \mathbf{q}, \mathcal{U}) \\ \widehat{\theta}_{\mathbf{d}} &= \arg\max_{\theta_D} p(\theta_D \,|\, \mathbf{d}, \mathcal{S}) \end{aligned}$$

Note that the factor $p(\widehat{\theta}_{\mathbf{d}} \,|\, \mathbf{d}, \mathcal{S})$ includes prior information about the document, and in general must be included when comparing the risk for different documents. This is critical when incorporating query-independent link analysis, or other extrinsic knowledge about a document. But, if we further assume that $p(\widehat{\theta}_{\mathbf{d}} \mid \mathbf{d}, \mathcal{S})$ is the same for all $\mathbf{d}$, so does not affect ranking, we will have the following very general distance-based (or equivalently, similarity-based) probabilistic model:

$$R(\mathbf{d};\mathbf{q}) \;\propto\; \Delta(\widehat{\theta}_{\mathbf{d}}, \widehat{\theta}_{\mathbf{q}})$$

We can view the vector space model as a special case of this general similarity model, where $\widehat{\theta}_{\mathbf{q}}$ and $\widehat{\theta}_{\mathbf{d}}$ are simply term vector parameters estimated heuristically and the distance function is the cosine or inner product measure.

We now consider a specific similarity model as an interesting special case, where θ_Q and θ_D are the parameters of unigram language models (i.e., $p(\cdot \mid \theta)$ is a distribution over a fixed word vocabulary), and the similarity measure is the Kullback-Leibler divergence

$$\Delta(\theta_Q, \theta_D) \;=\; \sum_w p(w \mid \theta_Q) \log \frac{p(w \mid \theta_Q)}{p(w \mid \theta_D)}$$

In this case

$$R(\mathbf{d};\mathbf{q}) \;\propto\; -\sum_w p(w \mid \widehat{\theta}_{\mathbf{q}}) \log p(w \mid \widehat{\theta}_{\mathbf{d}}) + c_{\mathbf{q}} \qquad (3)$$

where $c_{\mathbf{q}}$ is a constant that doesn't depend on the document, and so doesn't affect the retrieval performance.

According to this risk formula, the retrieval problem is essentially that of estimating $\widehat{\theta}_{\mathbf{q}}$ and $\widehat{\theta}_{\mathbf{d}}$. If $\widehat{\theta}_{\mathbf{q}}$ is just the empirical distribution of the query $\mathbf{q} = q_1 q_2 ... q_m$; that is,

$$p(w \mid \widehat{\theta}_{\mathbf{q}}) = -\frac{1}{m} \sum_{i=1}^{m} \delta(w, q_i)$$

where, δ is the indicator function, then we obtain

$$R(\mathbf{d};\mathbf{q}) \;\propto\; -\frac{1}{m} \sum_{i=1}^{m} \log p(q_i \mid \widehat{\theta}_{\mathbf{d}}) + c_{\mathbf{q}}$$

This is precisely the log-likelihood criterion that has been in used in all work on the language modeling approach to date. In the remainder of this paper we will develop new query expansion methods to estimate a model $\widehat{\theta}_{\mathbf{q}}$, and demonstrate that this model performs significantly better than using the empirical distribution for $\widehat{\theta}_{\mathbf{q}}$ when we use (3) as the risk.

4. MARKOV CHAINS FOR EXPANDING LANGUAGE MODELS

In this section we describe a Markov chain method for expanding language models for queries and documents, to be used in the formal framework just described. We begin by motivating the method in the context of translation models. We then explain the basic method and provide examples. In Section 7 we discuss the relationship between the Markov chain method and other techniques such as link analysis.

4.1 Markov chains on words and documents

As noted in Section 2, the translation models of Berger and Lafferty [1] can significantly improve retrieval performance, but must be estimated from training data. Since the parameters are highly lexical, an enormous amount of training data would be required to estimate them. Because the translation models are context-independent their ability to handle word sense ambiguity is limited. Moreover, the use of translation models for documents incurs a severe price in the time to score documents. The Markov chain method helps to overcome these limitations of the translation model paradigm.

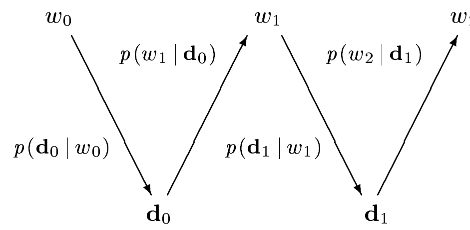

Figure 2: The Markov chain alternates between words and documents. For a given document, a word is selected according to the document language model. For a given word, a document is selected according to the posterior probability.

Our goal is to estimate a query model $\widehat{\theta}_{\mathbf{q}}$. For this purpose, we will estimate a probability $t(q \mid w)$ that a word w "translates" to the query term q. Imagine a user looking to formulate a query for an information need, and suppose, fancifully, that the user has an index available for the text collection to be searched. The user "surfs" the index in the following random manner. First, a word w_0 is chosen. Next, the index is consulted, and a document $\mathbf{d}_0$ containing that word is chosen. This choice will be influenced by the number of times the word appears in $\mathbf{d}_0$, and might be also affected by extrinsic data about the document, such as information about its author. From that document, a new word w_1 is sampled, a new document containing w_1 is chosen, and the process continues in this manner. After each step, there is some chance the user will stop browsing, as the topics of the documents drift further from the information need.

We now describe the random walk more precisely. The walk begins by choosing a word w_0 with probability $p(w_0 \mid \mathcal{U})$. At the i-th step, the user has selected word w_i. The user continues the random walk with probability α, generating a new document and word. With probability $1-\alpha$, the walk stops with word w_i. If the walk continues, then a document $\mathbf{d}_i$ is sampled from the inverse list for w_i, according to the posterior probability

$$p(\mathbf{d}_i \mid \boldsymbol{w}_i) \;=\; \frac{p(w_i \mid \mathbf{d}_i)\, p(\mathbf{d}_i)}{\sum_{\mathbf{d}} p(w_i \mid \mathbf{d})\, p(\mathbf{d})} \qquad (4)$$

where $p(\cdot \mid \mathbf{d})$ is the document language model, and where $p(\mathbf{d})$ is a prior distribution on documents. For example, with hypertext, $p(\mathbf{d})$ might be the distribution calculated using the "PageRank" scheme [4]. Having chosen a document $\mathbf{d}_i$, a new word w_{i+1} is sampled from it according to $p(\cdot \mid \mathbf{d}_i)$.

4.2 Matrix formulation

The algorithm can be cleanly described and related to other techniques using matrix notation. Let N be the number of terms in the word vocabulary, and M the number of documents in the collection. Let A be the $N \times M$ stochastic matrix with entries $A_{w,\mathbf{d}} = p(\mathbf{d} \mid w)$, where the probability is calculated as in equation (4). Also, let B be the $M \times N$ stochastic matrix with entries $B_{\mathbf{d},w} = p(w \mid \mathbf{d})$ given by the document language models. Finally, let C be the $N \times N$ stochastic matrix $C = A\,B$.

The probability that the chain stops after k steps with word $w_k = q$ is given by

$$(1-\alpha)\,\alpha^k\, C^k_{w,q}$$

where $C^k_{w,q}$ is the (w,q)-entry of the matrix C^k. Therefore, the overall probability of generating a word q is given by

$$\begin{aligned} t_\alpha(q \mid w) &= (1-\alpha)\left(I + \alpha\, C + \cdots \alpha^k\, C^k + \cdots\right)_{w,q} \\ &= (1-\alpha)\ (I - \alpha\, C)^{-1}_{w,q} \end{aligned}$$

Note that the matrix inverse $(I - \alpha\, C)^{-1}$ exists since, as a stochastic matrix, $\alpha^{-1} > 1$ cannot be an eigenvalue of C. We define this to be the translation probability of mapping w to q.

In the same way, we can calculate the probability that the user stops with document $\mathbf{d}$ as

$$\begin{aligned} t_\alpha(\mathbf{d} \mid w) &= (1-\alpha)\left[\left(I + \alpha\, C + \cdots \alpha^k\, C^k + \cdots\right) A\right]_{w,\mathbf{d}} \\ &= (1-\alpha)\ \left[(I - \alpha\, C)^{-1}\, A\right]_{w,\mathbf{d}} \end{aligned}$$

While the matrices A and B are sparse, so that the matrix product $C = A\,B$ can be computed efficiently, C^k quickly becomes dense as k increases, and the powers cannot be computed efficiently. However, as k increases, the "topic" wanders from the initial term w_0, as the probability quickly spreads out over all terms. Thus, intuitively, the first few steps of the chain are most important for retrieval purposes.

4.3 Expanding query and document models

Suppose that $\mathbf{q} = (q_1, q_2, \ldots, q_m)$ is a query that we wish to expand. In our framework, this means that we estimate a language model $p(w \mid \widehat{\theta}_{\mathbf{q}})$. Using the Markov chain method, this is done by calculating the posterior probability of words, according to the translation model for generating the query and a prior distribution on initial terms selected by the user. Thus, assuming the query terms are generated independently,

$$p(w \mid \widehat{\theta}_{\mathbf{q}}) \;\; \propto \;\; \sum_{i=1}^{m} t_\alpha(q_i \mid w)\, p(w \mid \mathcal{U})$$

A document $\mathbf{d}$ can be expanded using the Markov chain in a similar way:

$$p(w \mid \widehat{\theta}_{\mathbf{d}}) \;\; \propto \;\; t_\alpha(\mathbf{d} \mid w)\, p(w \mid \mathcal{U})$$

To understand how this method should be expected to work, it helps to consider running the chain for only one step. The probability of generating a query term q_i starting from an initial word w is equal to $p(w \mid \mathcal{U}) \sum_{\mathbf{d}} p(q_i \mid \mathbf{d})\, p(\mathbf{d} \mid w)$ in this case. The effect of the probability $p(\mathbf{d} \mid w)$ is similar to IDF in traditional retrieval methods, since this probability will be high if the word w appears in only a few documents. In particular, function words will typically appear in a very large number of documents, so $p(\mathbf{d} \mid w)$ will tend to be very small for such words. At the other end of the spectrum, if w appears *only* in the document $\mathbf{d}$, this probability will be one. Words with very high $p(\mathbf{d} \mid w)$ tend to be rare and specialized, and not sufficiently general to be useful to improve the language models. However, the prior $p(w \mid \mathcal{U})$ acts to select the more useful and frequent words having high IDF. At the same time, this prior gives a probabilistic mechanism for incorporating stop-word lists, or other extrinsic knowledge about the retrieval and query generation process. Thus, even the one-step chain captures many of the desirable features of term weighting schemes in a probabilistic model.

4.4 Incorporating feedback

Because the Markov chain translation probabilities $t(q \mid w)$ generate the query, the resulting expansion model $p(w \mid \mathbf{q})$ is fairly general. If query terms have multiple senses, a mixture of these senses may be present in the expanded model (See Figure 3). For semantic smoothing, a more context-dependent model that takes into account the relationship between query terms may be desirable. One way to accomplish this is through a pseudo-feedback mechanism. Suppose that a set of documents $\mathcal{D}(\mathbf{q})$ is known (or assumed) to be relevant to a query $\mathbf{q}$. We can condition the Markov chain to pass through this set. For example, in the one-step version of the random walk, we compute

$$p(w \mid q, \mathcal{D}(\mathbf{q})) \;\propto\; p(w) \sum_{\mathbf{d} \in \mathcal{D}(\mathbf{q})} p(\mathbf{d} \mid w)\, p(q \mid \mathbf{d})$$

In this way the expanded language model may be more "semantically coherent," capturing the topic implicit in the set of documents $\mathcal{D}(\mathbf{q})$ rather than representing words related to the query terms q_i in general. An example of this is shown in Figure 3. In Section 6 we report on experiments on TREC data that clearly demonstrate how this method can improve retrieval performance.

5. INDEXING SCHEMES

Calculation of the Markov chain probabilities on inverted indices manipulates document language models $p(w \mid \mathbf{d})$ and their posterior probabilities $p(\mathbf{d} \mid w)$. Computing these probabilities at retrieval time can be very expensive, but can be made more efficient by calculation of various statistics at index time, as discussed briefly in this section. This sheds further light on the role of document priors.

Standard indexing schemes store, for each index term w, a list of document indices in which the term appears, $\mathbf{d}_1 \mapsto \mathbf{d}_2 \mapsto \ldots \mapsto \mathbf{d}_{n(w)}$. For information retrieval based on language modeling, we require in addition the number of times the term appears in the document, and thus store a list $(\mathbf{d}_1, c(w, \mathbf{d}_1)) \mapsto (\mathbf{d}_2, c(w, \mathbf{d}_2)) \mapsto \cdots \mapsto (\mathbf{d}_{n(w)}, c(w, \mathbf{d}_{n(w)}))$. To normalize the language probabilities, at index time we compute the total document count $c(\mathbf{d}) = \sum_w c(w, \mathbf{d})$, and store this in a document array, together with a document prior $p(\mathbf{d})$.

In our implementation, we make two passes through the corpus. In the first pass a term vocabulary, document vocabulary, and document counts $c(\mathbf{d})$ are tabulated. In the second pass, the inverted lists and word marginals $p(w)$ are computed. Both the word-document and document-word lists are compressed using the γ-method [20].

To normalize the posterior probabilities, we must calculate $\sum_{\mathbf{d}} p(w \mid \mathbf{d})\, p(\mathbf{d})$. Using the maximum likelihood lan-

w	$p(w \mid \mathbf{q})$
virus	0.275
ebola	0.197
hoax	0.051
viruses	0.034
outbreak	0.034
fever	0.033
disease	0.024
haemorrhagic	0.023
gabon	0.022
infected	0.019
aids	0.016
security	0.014
monkeys	0.013
hiv	0.011
zaire	0.011

$\mathbf{q} =$ `ebola virus` (Web)

w	$p(w \mid \mathbf{q})$
star	0.361
wars	0.217
rpg	0.058
trek	0.033
starwars	0.032
movie	0.023
episode	0.020
movies	0.015
war	0.014
character	0.013
tv	0.013
film	0.012
fan	0.012
reviews	0.012
jedi	0.008

$\mathbf{q} =$ `star wars` (Web)

w	$p(w \mid \mathbf{q})$
star	0.192
wars	0.137
soviet	0.025
weapons	0.023
photo	0.020
army	0.020
armed	0.020
film	0.018
show	0.018
nations	0.017
strategic	0.017
tv	0.017
sunday	0.016
bush	0.014
series	0.013

$\mathbf{q} =$ `star wars` (TREC)

w	$p(w \mid \mathbf{q})$
star	0.170
wars	0.161
senate	0.069
strategic	0.050
spending	0.045
initiative	0.039
funding	0.036
vote	0.036
missile	0.033
billion	0.033
weapons	0.031
cheney	0.030
space	0.028
voted	0.021
missiles	0.020

$\mathbf{q} =$ `star wars` (TREC) with feedback

Figure 3: Sample query model probabilities using the Markov chain method. As seen in the third table, the probabilities can be fairly general and include a mixture of topics. Using the feedback approach, conditioning the chain on a document set obtained in a first pass, the query probabilities become more specialized, as seen in the fourth table.

guage model, this is given by $p(w) = \sum_{\mathbf{d}} \frac{c(w,\mathbf{d})}{c(\mathbf{d})} p(\mathbf{d})$. The choice of prior affects the indexing. With a uniform document prior $p(\mathbf{d}) = \frac{1}{|\mathcal{C}|}$, where $|\mathcal{C}|$ is the number of documents, we store

$$p(w) = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{d}} \frac{c(w,\mathbf{d})}{c(\mathbf{d})}$$

If one chooses a (rather unmotivated) prior where a document's probability is proportional to its total count $c(\mathbf{d})$ (so a user browses by favoring long documents), then this leads to $p(w) = \frac{c(w)}{\sum_w c(w)}$, which is simply the corpus unigram model.

Note that in our framework the prior information about a document enters in two places—in the Markov chain analysis of translation models, and in calculating the risk. We expect that significant improvements in query models can be obtained by basing the document prior on link-analysis or higher-level knowledge about the document collection; this remains an interesting topic for further research.

6. EXPERIMENTAL RESULTS

We evaluated the query model estimation methods described in the previous sections using three different TREC testing collections: the AP collection on disk 1 (topics 1–50), the TREC8 ad hoc task collection (topics 401–450 on disks 4&5 – CR), and the TREC8 web track collection (topics 401–450 on Web data). These data sets are representative of different aspects of TREC. The first is one of the earliest collections used in TREC and has relatively complete relevance judgments. The last two were selected because they represent relatively large collections and several published results on them are available. The Web data was also selected because of its unique document style.

When selecting queries, we set up our evaluation as an approximation of real world retrieval. Since queries are typically short, we only used the titles of each TREC topic description. The titles have an average length of 2.5 words, and typically contain one to four words each. The document collections were pre-indexed using the approach described in Section 5. All documents and queries were tokenized using the Porter stemmer. However, no stopword list was used, in order to test the robustness of our modeling techniques.

6.1 Effect of the query model

The query model obtained using the Markov chain for expansion is expected to perform better than the simple language model predicting the query, as in the Ponte-Croft work. In order to test this, we compared the retrieval performance of the original maximum likelihood query model with that of the query translation model on all three collections. The results are shown in Table 1. The figures for each model use the best choice of the document smoothing parameter λ in equation (1), as determined by a simple line search. In addition to non-interpolated average precision, which we consider to be the main performance measure, we include recall at 1,000 documents and initial precision, i.e., interpolated precision at 0% recall.

Compared with the simple query model, the basic query translation model improves average precision and recall significantly and consistently across all three collections. However, using the Markov chain with a seed set of 50 documents, similar to pseudo-feedback, as described in Section 4.4, gives a much greater improvement. For these experiments we use $\alpha = \frac{1}{2}$ and run the Markov chain for only two steps. The precision-recall curves are shown in Figure 4 for all six runs. In Figure 5, we compare the precision/recall of the basic query model and the expanded model at different settings of the document smoothing parameter λ. The figures clearly show that the query translation model is better than the simple query model for all settings of λ, and that the improvement is fairly insensitive to the choice of this parameter.

6.2 Query translation models vs. TF-IDF

The effect of query expansion using the Markov chain translation model should be compared with query expansion in more traditional retrieval models, such as the vector space model. For this purpose, we implemented a vector space model where the similarity is computed using the dot product and the TF formula is the well-known Okapi TF [19]. While the Okapi TF is designed to be used in the

Collection		Simple LM	Query Model	Improv.	QM w/ Pseudo	Improv.
AP89	AvgPr	0.188	0.201	+7%	0.232	+23%
	InitPr	0.515	0.500	−3%	0.534	+4%
	Recall	1510/3261	1745/3261	+16%	2019/3261	+34%
TREC8	AvgPr	0.241	0.266	+10%	0.294	+22%
	InitPr	0.620	0.723	+17%	0.676	+9%
	Recall	2791/4728	2913/4728	+4%	3368/4728	+21%
WEB	AvgPr	0.244	0.275	+13%	0.304	+25%
	InitPr	0.607	0.664	+9%	0.663	+9%
	Recall	1760/2279	1848/2279	+5%	1910/2279	+9%

Table 1: Comparison of the basic language modeling method with expanded query models. Column three gives the performance using the Markov chain query translation model; column 5 shows the effect of including an initial document set (pseudo-feedback) to condition the Markov chain.

Collection		TF-IDF+Rocchio	Query Model	Improv.	QM w/ Pseudo	Improv.
AP89	AvgPr	0.230	0.201	−13%	0.232	+1%
	InitPr	0.492	0.500	+2%	0.534	+9%
	Recall	2082/3261	1745/3261	−16%	2019/3261	−3%
TREC8	AvgPr	0.256	0.266	+4%	0.294	+15%
	InitPr	0.637	0.723	+14%	0.676	+6%
	Recall	3154/4728	2913/4728	−8%	3368/4728	+7%
WEB	AvgPr	0.226	0.275	+22%	0.304	+35%
	InitPr	0.559	0.664	+19%	0.663	+19%
	Recall	1729/2279	1848/2279	+7%	1910/2279	+10%

Table 2: Comparison of TF-IDF with Rocchio to Markov chain query expansion in the language modeling framework.

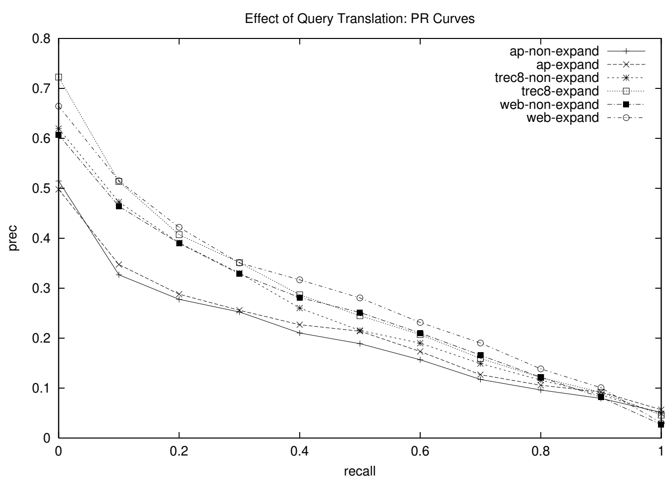

Figure 4: Precision-recall curves for all three collections, comparing the simple language model to the query translation model.

BM25 retrieval function, in practice, we have found that using it in the vector space model also tends to give very good performance. For query expansion, we implemented a simplified Rocchio, where we add only positive terms to the query, controlled by three parameters: (1) the number of documents for blind feedback, (2) the number of terms to add, and (3) the relative coefficient of the added terms. We varied these parameters among several values and chose the best performing parameters for comparison.

It is interesting to see that the relative performance of the query translation model and the TF-IDF model varies from collection to collection. Constraining the Markov chain to use a selected set of documents, obtained during a first retrieval pass, as described in Section 4, generally gives the best performance. However, on AP89, the performance of Rocchio and the query translation model are virtually the same. The greatest gain from the query translation model comes on the Web data, where the query models achieve a 35% improvement over Rocchio. We note that, although we use only the title queries, which are very short, our results on both the TREC8 and Web data using query models are quite comparable to the official TREC submissions, which use the full queries.

7. RELATED WORK

There is a large and rich literature on probabilistic models in information retrieval, and it would not be possible to survey it here. The work presented in this paper is most closely related to recent developments in the language modeling approach [17, 11, 15, 1]. Important precursors to the language modeling approach include [3, 18, 8, 10, 21]. The framework based on risk minimization that we have introduced is very natural and general. We are not aware of any directly comparable framework in the IR literature, although several early papers discuss indexing schemes designed to optimize utility measures [16, 7, 2]. The approach that we present differs significantly from this work in that the central components are probabilistic models of documents and queries, combined using an explicit loss function according to Bayesian decision theory. The formal model is meant to explicitly represent the uncertainty inherent in our model of

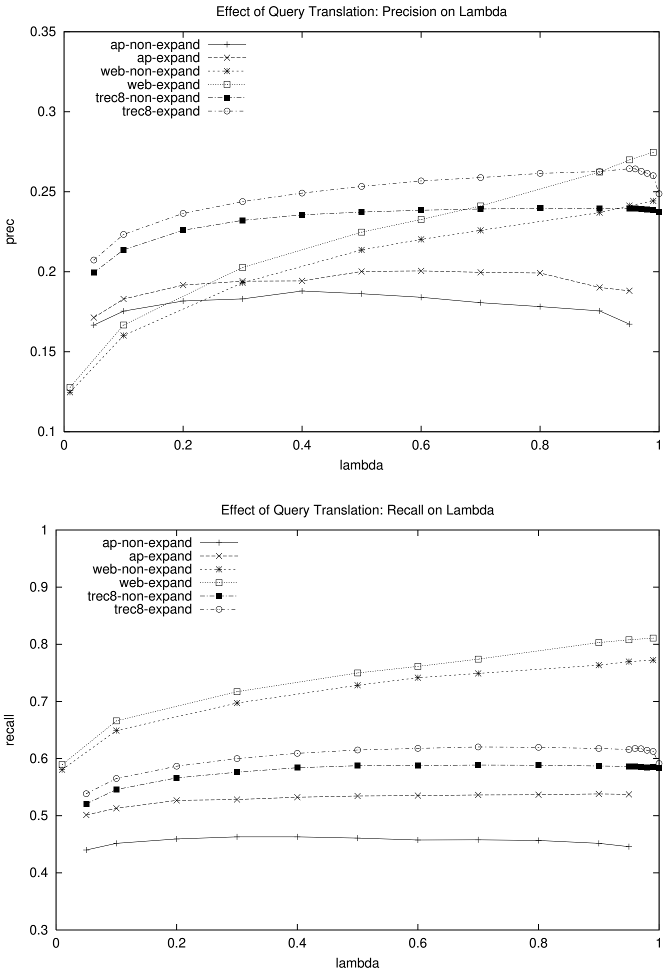

Figure 5: Effect of λ, the document smoothing parameter for linear interpolation. The results indicate that the improvement of the query translation model over the simple query model is fairly insensitive to the choice of λ.

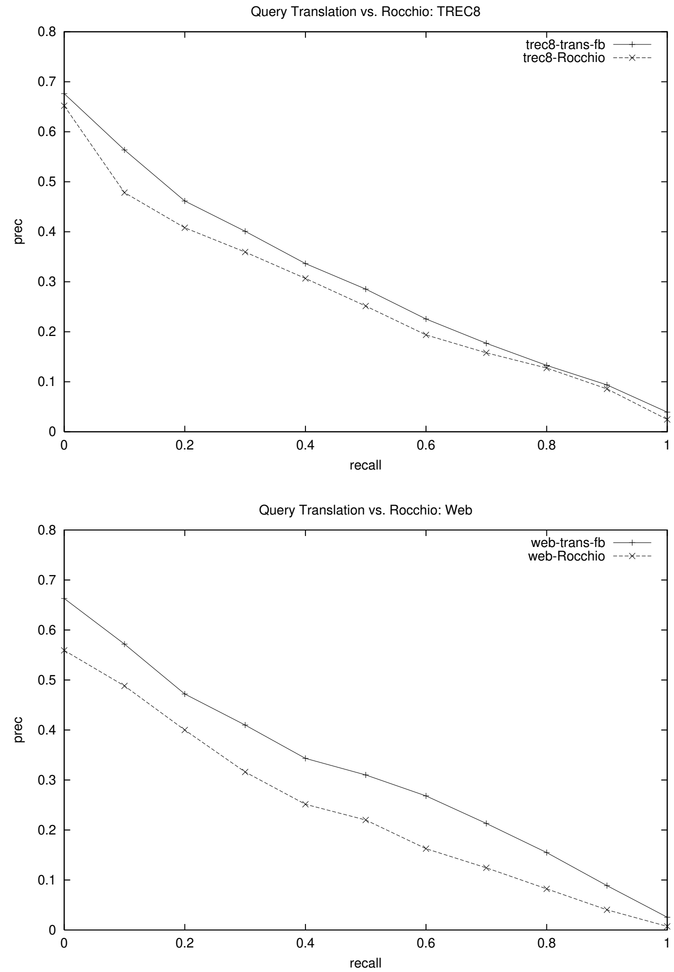

Figure 6: Precision-recall curves comparing TF-IDF with Rocchio to the query translation models. The curves for the two methods on AP89 are nearly the same, and so are not shown.

the user and collection, and is an attractive way in which to think about query and document language models. An interesting direction for future work will be to go beyond the use of a single document and query model (as in MAP estimation).

The practical implementation of our new query expansion technique involves using only a small number of steps of a Markov chain on inverted indices. However, to compare this technique to previous work, it is best to imagine using the full matrix analysis, which involves computing the matrix inverse $(I-\alpha C)^{-1}$, as described in Section 4. When viewed in this way, there are interesting connections between the Markov chain approach to building query models, link analysis methods, theory of social networks, and latent semantic indexing.

Methods based on latent semantic analysis (LSA) [9] work with the singular value decomposition of the word-document matrix $\widehat{A}$ having entries $\widehat{A}_{w,\mathbf{d}} = c(w,\mathbf{d})$. Let $\widehat{B}$ denote the transpose $\widehat{A}^{\top}$, so that $\widehat{B}_{\mathbf{d},w} = c(w,\mathbf{d})$. LSA computes projections onto eigenspaces of the matrices $\widehat{A}\,\widehat{B}$ and $\widehat{B}\,\widehat{A}$, building a low-dimensional subspace to represent terms and documents. In our method, we work with the closely related matrices A and B. The matrix B is obtained from $\widehat{B}$ by normalizing the rows. However, the matrix of posterior probabilities A is obtained from $\widehat{A}$ by normalizing the rows only in the case of the rather unnatural document prior $p(\mathbf{d}) \propto c(\mathbf{d})$. But the essential difference is that our method interprets the matrix $(I-\alpha AB)^{-1}$ probabilistically, rather than using a vector space approach that projects onto subspaces generated by the top eigenvectors of $\widehat{A}\,\widehat{B}$.

There is a closer connection to methods from link and citation analysis. For example, Kleinberg's "hubs and authorities" technique [13] uses an initial document set $\mathcal{D}(\mathbf{q})$ for a query, defines the matrix $\widehat{B}$ encoding all outgoing links from $\mathcal{D}$, and the matrix $\widehat{A}$ encoding all incoming links to $\mathcal{D}$. The "hub score" of a document is then defined in terms of the principal eigenvector of the matrix $\widehat{A}\,\widehat{B}$ (ignoring some details involving normalization). To cast this in terms of our query expansion method, forward links are replaced by document word indices, with language model probabilities $p(w \mid \mathbf{d}) = B_{\mathbf{d},w}$, and incoming links are replaced by inverted file indices, with posterior probabilities $p(\mathbf{d} \mid w) = A_{w,\mathbf{d}}$. Use of the principal eigenvector of the matrix AB could give an excellent method for query expansion.

Instead, we have chosen to use the Markov chain leading to the matrix $(I - \alpha AB)^{-1}$, which we believe gives greater flexibility, as well as numerical stability through smoothing. Intuitively, the first iterations of the walk are most important, and they are emphasized using the parameter α. The use of this matrix is related to research on social networks carried out nearly 50 years ago; an excellent discussion of these methods is given by Kleinberg [13]. To estimate the "standing" of an individual in a social network, Katz [12] uses a matrix C where $C_{i,j}$ is the strength of an "endorsement" of individual j by individual i, and defines the standing of an individual j as the j-th column of the matrix $(I - \alpha C)^{-1} - I$. Very similar measures are defined by Hubbel [5].

8. SUMMARY AND CONCLUSIONS

We have presented a new framework for information retrieval based on Bayesian decision theory. In this framework we assume a probabilistic model for the parameters of document and query language models, and cast the retrieval problem in terms of risk minimization. The framework is very general and expressive, and by choosing specific models and loss functions it is possible to recover many previously developed frameworks. In particular, previous approaches based on language modeling and query-likelihood ranking are obtained as a natural special case. In this paper we focus on the use of Kullback-Leibler divergence as loss function, and the estimation of query language models. We introduce a novel method for estimating query models that uses Markov chains on the inverted indices of a document collection. This random walk has a natural interpretation in terms of document language models, and results in practical and effective translation models and query language models. Experiments on standard TREC methods indicate the usefulness of both the framework and the Markov chain method, as we obtain significant improvements over standard query expansion methods for strong baseline TF-IDF methods, with the greatest improvements attained for short queries on Web data.

ACKNOWLEDGEMENTS

We thank Jamie Callan, Bruce Croft, Imre Kondor, Victor Lavrenko, Guy Lebanon, and Roni Rosenfeld for valuable discussions related to this work. This research was sponsored in part by the Advanced Research and Development Activity in Information Technology (ARDA) under its Statistical Language Modeling for Information Retrieval Research Program.

REFERENCES

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

[2] A. Bookstein and D. Swanson. A decision theoretic foundation for indexing. *Journal for the American Society for Information Science*, pages 45–50, 1975.

[3] A. Bookstein and D. Swanson. Probabilistic models for automatic indexing. *Journal for the American Society for Information Science*, 25(5):312–318, 1976.

[4] S. Brin and L. Page. Anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

[5] H. Hubbell C. An input-output approach to clique identification. *Sociometry*, 28:377–399, 1965.

[6] J. G. Carbonell, Y. Geng, and J. Goldstein. Automated query-relevant summarization and diversity-based reranking. In *IJCAI-97 Workshop on AI and Digital Libraries*, 1997.

[7] W. S. Cooper and M. E. Maron. Foundations of probabilistic and utility-theoretic indexing. *Journal of the Association for Computing Machinery*, 25(1):67–80, 1978.

[8] W. B. Croft and D.J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35:285–295, 1979.

[9] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of American Society for Information Science*, 41:391–407, 1990.

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

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

[12] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

[13] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the Association for Computing Machinery*, 46, 1999.

[14] J. Lafferty and C. Zhai. Probabilistic IR models based on query and document generation. In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, Carnegie Mellon University, May 31–June 1, 2001.

[15] D. H. Miller, T. Leek, and R. Schwartz. 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, 1999.

[16] F. Mosteller and D. Wallace. *Inference and disputed authorship: The Federalist.* Addison Wesley, 1964.

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

[18] S. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[19] S. E. Robertson, S. Walker, S. Jones, M. M.Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In D. K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, 1995.

[20] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Morgan Kaufmann, 1999.

[21] S. K. M. Wong and Y. Y. Yao. A probability distribution model for information retrieval. *Information Processing and Management*, 25(1):39–53, 1989.