Bayesian Information Retrieval: Preliminary Evaluation*

Michelle Keim* **David D. Lewis†** **David Madigan***

*Dept. of Statistics, Box 354322
University of Washington
Seattle, WA 98195
keim,madigan@stat.washington.edu

†AT&T Labs – Research
600 Mountain Ave., 2A-410
Murray Hill, NJ 07974
lewis@research.att.com

Abstract

Given a database of documents and a user's query, how can we locate those documents that meet the user's information needs? Because there is no precise definition of which documents in the database match the user's query, uncertainty is inherent in the information retrieval process. Therefore, probability theory is a natural tool for formalizing the retrieval task. In this paper, we propose a Bayesian approach to one of the conventional probabilistic information retrieval models. We discuss the motivation for such a model, describe its implementation, and present some experimental results.

1 Introduction

Information Retrieval (IR) is a branch of computer science that deals with automated information storage and retrieval. The goal of a text retrieval system is to find those documents in a text database that are useful for a specific user's needs. We say that a document is *relevant* if a user finds it useful in answering his/her query, otherwise the document is *non-relevant*. Information retrieval systems differ from conventional database systems in that there is no precise definition of which elements in the document collection match the user's query. Because of the uncertainty inherent in the retrieval process, probability theory is a natural tool for formalizing the retrieval process.

In this paper, we begin by briefly introducing information retrieval and describing the probabilistic model upon which our approach is based. We then discuss the weaknesses of the existing model and the motivation for a Bayesian model. Next, we describe the Bayesian model and some of the details in implementing it. We conclude with a presentation of some preliminary results and our plans for anticipated future work.

*A grant from the National Cancer Institute to Mathsoft Inc. and the University of Washington supported this work.

2 A Probabilistic IR Model

Probabilistic information retrieval models date back to the early 60's (Maron and Kuhns 1960), but have rarely been used in operational retrieval systems. However, the probabilistic model was perhaps the first IR model with a firm theoretical foundation. The goal of a probabilistic model is to estimate $P_q(R \mid d_k)$, the probability that adocument, d_k, is relevant (R) to a query, q.

The probabilistic model that is discussed most often in the IR literature is the *binary independence model.* Robertson and Sparck Jones (1976) first introduced this model in the context of relevance weighting, and van Rijsbergen (1979) also provides a thorough description of this model.

In order to compute $P_q(R \mid d_k)$, an information retrieval system must somehow represent and store the documents. IR systems frequently represent a document by a set of words known as *index terms.* In general, index terms are those words in the document that remain after the words on a "stop list" (a list of common words) are stripped out. These words are often "stemmed" by removing prefixes and suffixes. We can then represent documents by a vector

$$\mathbf{t} = (t_1, t_2, \ldots, t_p)$$

where p is the number of index terms. In the binary independence model, the values of the vector elements, t_i, are binary, indicating the presence or absence of the term in the documents. More generally, they may be counts or weights which indicate the importance of the term in the document.

With this document representation, we can use Bayes'

Rule to express the probability that a document, represented by $\mathbf{t} = (t_1, t_2, \ldots, t_p)$, is relevant to a specific query, q as

$$P_q(R|\mathbf{t}) \propto P_q(\mathbf{t}|R)P_q(R).$$

If we assume that the terms are conditionally independent given relevance and non-relevance, we obtain a simple expression for the log odds of relevance, according to which we can rank the documents:

$$\log \frac{P_q(R|\mathbf{t})}{P_q(\bar{R}|\mathbf{t})} = \log \frac{P_q(R)}{P_q(\bar{R})} + \sum_{i=1}^{p} \log \frac{P_q(t_i|R)}{P_q(t_i|\bar{R})}. \quad (1)$$

The summation is usually restricted to the terms that occur in the query, under the assumption that if a term is not in the query it is equally likely to be in a relevant document as in a non-relevant document. The term $\log \frac{P_q(R)}{P_q(\bar{R})}$ is constant for a particular query. Therefore, if we simply wish to present documents to the user in order of $\log \frac{P_q(R|\mathbf{t})}{P_q(\bar{R}|\mathbf{t})}$, without displaying the actual values, we can ignore $\log \frac{P_q(R)}{P_q(\bar{R})}$. Then, in order to apply this model, we require estimates for $P_q(t_i = 1|R)$ and $P_q(t_i = 1|\bar{R})$, for all terms in the query. That is, we need to estimate the probability that the term t_i occurs in a document that is relevant (or non-relevant) to the current query, q.

This estimation requires that we look at the frequency of term occurrences in the sets of relevant and non-relevant documents. However, at the outset, we do not know the status, relevant or non-relevant, for any of the documents. Probabilistic IR systems typically circumvent this problem by producing a initial ranked list of documents based on ad hoc estimation of the probabilistic model parameters or an alternate retrieval method (e.g. ranking the documents according to the number of index terms in common with the query). The system then presents the top-ranked documents to the user for judgment as to whether they are relevant or not. Based on this relevance data, it is possible to estimate the parameters of the probabilistic model by computing the proportion of times each term occurs in the documents that have been judged relevant and non-relevant. This process of obtaining relevance information from the user and using it in a further search is called *relevance feedback*.

One of the weakness of the probabilistic model is that it uses two different methods: one to produce the initial document ranking, and another after the user provides relevance judgments. In addition, the parameter estimates based on the relevance feedback data may be unreliable due to the typically small sample sizes of judged documents.

3 Motivation for Bayesian IR Model

Despite the attractiveness of an information retrieval model with firm theoretical foundations, the existing methods of estimating the parameters of the probabilistic model are somewhat unsatisfactory. Because the probabilistic model requires relevance judgments for the *current* query to estimate its parameters, it must resort to an alternative method to produce the intial document ranking. In addition, the conventional probabilistic model is unable to utilize all the data that is potentially available and useful. For instance, an IR system may accumulate a large database of relevance judgments for *past* queries. In regards to the probabilistic model, Fuhr (1992) states, "This model makes very poor use of the relevance feedback information given by the user, since this information is only considered in the ranking process for the current query. For a new query, none of this data can be used at all." If some of the queries issued in the past are similar to the current query, the relevance judgments obtained for these past queries might be useful in determining which documents are relevant to the current query.

Therefore, we seek a principled IR model with probabilistic foundations which overcomes some of the weaknesses of existing probabilistic models. A Bayesian approach to the probabilistic model accomplishes this and has the following strengths:

1. A Bayesian approach to the probabilistic model retains the sound theoretical basis of the traditional probabilistic models, but is able to produce an initial document ranking without relying on alternate retrieval methods or ad hoc considerations.

2. Relevance feedback fits naturally into the model. A Bayesian approach provides an automatic mechanism for learning. By placing prior *distributions* on the model parameters, we can coherently update these distributions as more feedback data become available. Thus, the Bayesian approach uses the same modle both before and after relevance feedback data is available.

3. A Bayesian approach allows us to incorporate relevance information from *other queries*. It provides a better framework for making use of knowledge about queries. We can use relevance feedback data from past queries to place informative prior distributions on the model parameters. We will incorporate ad-hoc judgments about inter-query similarities and term importance into these prior distributions.

4 Bayesian IR Model

While we could apply a Bayesian approach to a wide variety of IR models, we focus on the binary independence model discussed in Section 2. Gelman et al. (1995) is a good introduction to Bayesian analysis. Recall that the parameters of our model are $P_q(t_i|R)$ and $P_q(t_i|\bar{R})$, $i = 1, \ldots, p$. Rather than using ad hoc techniques or relying on the availability of relevance judgments for the current query to obtain point estimates for $\pi_{R_i} = P_q(t_i|R)$ and $\pi_{\bar{R}_i} = P_q(t_i|\bar{R})$, we assess prior *distributions*, $p(\pi_{R_i})$ and $p(\pi_{\bar{R}_i})$, $i = 1, \ldots, p$, for these parameters. These prior distributions will embody any prior knowledge about query-document relationships. We conjecture that we may obtain this prior knowledge from the relevance data on *past queries*, the occurrence of terms in the current query, and the frequency of terms in the documents.

In order to compute the initial probabilities of relevance for each document, we use the expected value of the prior distributions to estimate the model parameters. Therefore, we base the initial document ranking solely on the prior distributions.

The system may then present documents to the user and solicit relevance judgments. These relevance judgments provide us with *query-specific* relevance data, $\mathbf{X_r} = (X_{r_1}, X_{r_2}, \ldots, X_{r_k})$, which we will use to update the distribution on the model parameters to obtain $p(\pi_{R_i}|\mathbf{X_r})$ and $p(\pi_{\bar{R}_i}|\mathbf{X_r})$, $i = 1, \ldots, p$. Thus, the system has the ability to "learn" as it interacts with the user.

Since we are assuming that the index terms are binary, when the user provides us with relevance judgments, we observe either the presence or absence of each term in each document in the set of relevant (and non-relevant) documents. Thus, binomial distibution is a natural model for the number of occurrences of term i in the set of relevant documents (which we denote by n_{R_i}):

$$p(n_{R_i}|\pi_{R_i}) = \binom{r_k}{n_{R_i}} (\pi_{R_i})^{n_{R_i}} (1 - \pi_{R_i})^{r_k - n_{R_i}}, \quad (2)$$

where r_k is the number of documents that the user has judged relevant.

We must now specify a prior distribution for the binomial probability, $\pi_{R_i} = P_q(t_i|R)$. The beta distribution is a conjugate with binomial sampling. *Conjugacy* is the property that the posterior distribution has the same form as the prior distribution. Because conjugate priors are mathematically convenient, we initially assume that each of the probabilities $\pi_{R_i} = P_q(t_i|R)$, $i = 1, \ldots, p$, has a Beta$(\alpha_{R_i}, \beta_{R_i})$ distribution. A beta distribution with parameters α and β has the form:

$$p(\pi_{R_i}) \propto (\pi_{R_i})^{\alpha-1}(1 - \pi_{R_i})^{\beta-1}. \quad (3)$$

Gelman et al. (1995), Chapter 2, provides a thorough description of the beta-binomial model.

The use of beta prior distributions is also attractive because it provides a simple and straightforward mechanism for learning. The updated distributions become

$$p(\pi_{R_i}|\mathbf{X_r}) \sim \text{Beta}(\alpha_{R_i} + a_{R_i}, \beta_{R_i} + b_{R_i}), \quad (4)$$

where a_{R_i} is the number of relevant documents in the sample that contain term i and b_{R_i} is the number of relevant documents in the sample that do not contain term i. The distribution in Equation 4 then becomes the prior distribution for the next iteration of retrieval. As the number of judged documents increases, we obtain more precise knowledge of the model parameters.

There are two main issues in implementing the Bayesian IR model:

1. *Specification of Prior Distributions:* How do we incorporate our prior knowledge about the model parameters in the prior distributions? In other words, how do we choose the parameters of the prior distributions, α_{R_i} and β_{R_i}?

2. *Updating the Distributions:* Which, and how many, documents should we present to the user for relevance feedback? What is the effect of the variance of each of the prior distributions on the retrieval performance in subsequent iterations?

5 Specification of Prior Distributions

5.1 Related Work

Other researchers' implementations of the probabilistic model often explicitly, or implicitly assume a beta distribution on the model parameters and they frequently choose $\alpha_{R_i} = \beta_{R_i} = 0.5$ for every i. This is the approach taken Croft and Harper (1979). In addition, Robertson and Sparck Jones (1976) suggest estimating $P_q(t_i = 1|R)$ by $\frac{a_{R_i}+0.5}{a_{R_i}+b_{R_i}+1}$, which is is equivalent to using a Beta(0.5,0.5) prior distribution.

Bookstein (1983) presents a *sequential learning model* which also utilizes Bayesian learning. However, he states that the choice of initial values for the parameters of the prior distribution is an open question. Losee (1988) followed up on Bookstein's model and experimented with some arbitrarily-set parameters for the prior distributions (e.g. $\{\alpha_{R_i}, \beta_{R_i}, \alpha_{\bar{R}_i}, \beta_{\bar{R}_i}\} = \{5,8,4,8\}$ for all terms). He also recommended using

Croft and Harper's estimates for $\alpha_{\bar{R}_i}$ and $\beta_{\bar{R}_i}$. Under the assumption that most of the documents in the database are non-relevant, Croft and Harper suggested estimating the probability that a term occurs in non-relevant documents based on its frequency of occurrence in the entire document collection. This idea gives rise to estimates of $\alpha_{\bar{R}_i}$ and $\beta_{\bar{R}_i}$ such that $\frac{\alpha_{\bar{R}_i}}{\beta_{\bar{R}_i}} = \frac{n_i}{N-n_i}$, where N is the number of documents in the collection and n_i is the number that contain term i. The values chosen for $\alpha_{\bar{R}_i}$ and $\beta_{\bar{R}_i}$ are scaled according to how confident one is about the estimates. Losee also suggested estimating the parameters α_{R_i} and β_{R_i} based on term occurrences in the set of documents in the database that have been judged relevant to *any* query. This results in a single estimate of $\{\alpha_{R_i}, \beta_{R_i}\}$ which is the same for every query.

5.2 Proposed Method of Specification

We suggest a method that better uses the query information in the database. In order to construct the prior distributions, we primarily utilize the available relevance data for past queries that are "similar" to the current query. There are several issues involved in incorporating this information into the prior distributions:

1. How do we determine which past queries are similar to the current query?
2. How do we combine the knowledge we have from each similar query?
3. How do we incorporate knowledge from other sources, such as the terms in the new query, into the prior distribution?

Finding Similar Queries Ideally, we would estimate $P_q(t_i = 1|R)$ and $P_q(t_i = 1|\bar{R})$ based on term occurrences in the documents that are relevant and non-relevant to the current query. Therefore, "similar" queries should be those past queries that have as many of the same relevant documents in common with the current query as possible. How can we identify those queries?

As far as we know, there is very little literature that investigates similarity among queries. Raghavan and Sever (1995) explore query similarity in order to find the "past optimal query" that is most similar to the user's current query. However, the similarity measures that they discuss require document rankings for the two queries being compared. Their methods are not applicable in our situation because initially we do not have a ranked list of documents for the new query.

Voorhees et al. (1995) discuss query similarity for a different application. They develop collection fusion strategies to successfully merge the results of retrieval runs on separate independent collections into a single result. They suggest two ways to determine similarity between queries. Their first method applies the vector space model (Salton 1971) to queries and their second method clusters queries based on the number of common relevant documents retrieved. We have adopted the vector space approach.

The vector space approach represents a query as a vector, where each vector element corresponds to a possible term. These elements may be binary, but they are usually term weights which represent the term's importance in the query. If the term is not in the query it gets a weight of zero. If the term is in the query its weight is often a function of how often it occurs in the query (term frequency) and how often it occurs in the document collection (collection frequency). We use a weight that is the product of the term frequency and the log of the inverse collection frequency. The query-to-query similarity is then just the inner product between the new query's term vector, $t_{Q_{new}}$, and the past query's term vector, $t_{Q_{past}}$:

$$sim(Q_{past}, Q_{new}) = t_{Q_{past}} \cdot t_{Q_{new}}. \quad (5)$$

We have also considered some variations on this similarity measure, but found that they did not significantly affect the model's performance. Whatever the similarity measure, we must chose a threshold above which we define queries to be "similar." We investigate this through empirical experimentation.

Combining Knowledge from Past Queries For each past query, Q_{past}, the availability of relevance judgments allows us to estimate $p_{Q_{past}}(t_i|R)$ and $p_{Q_{past}}(t_i|\bar{R})$. After we determine which past queries are similar to the user's current query, the next issue is how to use these estimates to specify the distribution of $p_{Q_{new}}(t_i|R)$ and $p_{Q_{new}}(t_i|\bar{R})$ for the new query.

For each past query, we assume that we have complete relevance judgments. This gives us the following data:

$$\begin{aligned} a_{ij} &= |R_{Q_j} \ni t_i = 1| \\ b_{ij} &= |R_{Q_j} \ni t_i = 0| \\ c_{ij} &= |\bar{R}_{Q_j} \ni t_i = 1| \\ d_{ij} &= |\bar{R}_{Q_j} \ni t_i = 0| \end{aligned}$$

where R_{Q_j} and $\bar{R}_{Q_j}$ are the sets of documents relevant and non-relevant to Q_j, respectively and $|\cdot|$ gives the size of the set. For example, a_{ij} is the number of documents that are relevant to query j and contain term i.

If there are m past queries in the database, $Q_1, \ldots, Q_m$, that are similar to the current query, then we have m estimates of $p_{Q_j}(t_i|R)$ and $p_{Q_j}(t_i|\bar{R})$:

$$\hat{p}_{Q_j}(t_i|R) = \frac{a_{ij}}{a_{ij}+b_{ij}}, \quad \hat{p}_{Q_j}(t_i|\bar{R}) = \frac{c_{ij}}{c_{ij}+d_{ij}}. \quad (6)$$

The estimates of $p_{Q_j}(t_i|R)$ for each of the past queries may be combined in two ways: *unweighted* and *weighted* combinations. By simply aggregating all of the information from the m similar queries without regard to their strength of similarity to the current query, we obtain an *unweighted* specification of α_i and β_i:

$$\alpha_i = \sum_{j=1}^{m} a_{ij}, \quad \beta_i = \sum_{j=1}^{m} b_{ij}. \quad (7)$$

Alternatively, we can account for the degree of similarity between the new query and each of the past queries and weight the information from each query accordingly. A *weighted* specification is

$$\alpha_i = \sum_{j=1}^{m} w_j a_{ij} \quad \beta_i = \sum_{j=1}^{m} w_j b_{ij}, \quad (8)$$

where $w_j = sim(Q_j, Q_{new})$. We found that weighting the information from the past queries improved the model's performance. In addition, when we weight the queries, performance does not significantly differ when we vary the choice of similarity measure and the threshold for similarity.

These methods provide us with an estimate of the ratio of α_i to β_i. We may then wish to revise the variance of the prior distribution by restricting $\alpha_i + \beta_i$ to be less than some constant. The variance specification is somewhat subjective and may reflect how confident we feel about our prior knowledge of $p_{Q_{new}}(t_i|R)$ based on the past queries. We investigate the optimum variance empirically.

Incorporating Query Term Information When specifying the prior distributions, we would like to incorporate all prior knowledge that we have about term occurrences in relevant and non-relevant documents. In addition to data from similar past queries, we also have some prior beliefs about terms that are in the user's query. If a term occurs in the query, there is a good chance that it will occur in documents that the user will find relevant. Therefore, $P_q(t_i|R)$ is likely to be large for query terms. We quantify "large" by referring to Croft and Harper (1979). They propose various ways to estimate the parameters of the probabilistic model when no relevance information is available. They experimented with setting $P_q(t_i|R)$ to a constant for all query terms and found that setting $P_q(t_i|R) = 0.9$ produced the best results. Therefore, we experimented with increasing α_{R_i} to reflect this knowledge.

Ranking the Documents Once we have specified the prior distributions, we substitute their means for $P_q(t_i|R)$ and $P_q(t_i|\bar{R})$ in Equation 1. Thus, we initially rank the documents according to:

$$\sum_{\{i:t_i=1\}} \log \frac{\frac{\alpha_{R_i}}{\alpha_{R_i}+\beta_{R_i}}}{\frac{\alpha_{\bar{R}_i}}{\alpha_{\bar{R}_i}+\beta_{\bar{R}_i}}} + \sum_{\{i:t_i=0\}} \log \frac{\frac{\beta_{R_i}}{\alpha_{R_i}+\beta_{R_i}}}{\frac{\beta_{\bar{R}_i}}{\alpha_{\bar{R}_i}+\beta_{\bar{R}_i}}} + C \quad (9)$$

where C is constant for a given query and both summations are over terms that occur in the current query.

5.3 Updating the Distributions

After we rank the documents, we present a subset of documents to the user and request relevance judgments. This provides us with query-specific data which we use to update the prior distributions. The updating simply requires that we add the observed counts to the appropriate parameter as in Equation 4. For example, suppose our prior distribution for $P_q(t_i|R)$ was Beta(2, 8). If the user judged 7 relevant documents, of which 4 contained term i, then the posterior distribution would become Beta(6, 11). We then re-rank the documents based on the updated distributions. Because we learned about the current query, we expect that the model will perform better in each successive iteration.

Although information retrieval systems typically solicit relevance judgments for the documents with the highest probability of relevance, it may be worthwhile to request relevance judgments on other documents as well. Lewis and Gale (1994) suggest that we may more effectively train a classifier if we chose for training those cases about which we are least certain of class membership. Perhaps we may improve retrieval performance in the long-run if we ask the user to label some documents in the middle of the ranked list.

In addition, the variance we choose for the prior distributions will affect how much the user's relevance judgments will influence the posterior distributions. This is another issue we investigate empirically.

6 Evaluation

IR researchers usually evaluate the effectiveness of a new retrieval method by testing the method on standard test collections. These test collections have been developed over the years for use in different retrieval experiments and consist of queries, documents, and relevance judgments. (See Sparck Jones and van Rijsbergen (1976) for details.) We have chosen to use the *Cranfield 1400* collection for preliminary experimentation. The Cranfield collection has 1400 documents and 225 queries. The documents have an average of

53 terms and the queries have an average of 9 terms. Each query has an average of 8 relevant documents.

Retrieval effectiveness is usually measured by two quantities: *precision* and *recall*. Precision is the proportion of retrieved documents that are actually relevant and recall is the proportion of the relevant documents that are retrieved. Thus, retrieval effectiveness is treated as the ability of the system to retrieve relevant documents while holding back non-relevant ones. van Rijsbergen (1979), Chapter 7, and Salton (1971), Chapter 5, explain the methodology for computing precision and recall values in testing situations. Typically a set of standard recall values are specified (usually 10%, 20%,..., 100%) for which we calculate corresponding precision values for each query. Then the precision values are averaged across the queries.

Evaluating retrieval effectiveness after the user has provided some relevance judgments can be misleading. If we re-rank the entire set of documents in light of the feedback data, the documents that the user previously ranked will likely move to the top of the ranking. This "ranking effect" can make the evaluation of the effect of relevance feedback appear better than it is, since most of the improvement comes from re-ranking the documents that the user has already seen. We would prefer to evaluate the "feedback effect," the improvement in performance due to the ranking of new, unseen documents. To do so, we use the "test and control method" (Salton 1971, Chapter 17).

We randomly split the document collection in half. We use the first half, the "test group," to run an initial search based on the prior distributions. We then obtain relevance judgments for some documents and update the distributions. Then, we use the second half, the "control group," to evaluate the feedback effect. To do this, we compute relevance probabilities for the documents in the control group based both on the prior distributions and based on the updated distributions. We can then compare these two runs without the interference of a ranking effect.

We randomly divided the Cranfield queries into two sets: one set of 100 queries which we treat as the training queries and one set of 125 queries which we treat as test queries. Table 1 summarizes the evaluation method.

7 Results

In this section, we present the results from two experiments. In the first experiment, we use only the data from similar past queries to specify the parameters of the prior distributions. We explored various similarity measures, thresholds for similarity, combinations of past query data, and variances. We then used the specification which produced the best results to investigate the effect of relevance feedback. In the second experiment, we incorporate knowledge about the query terms, as well as past query data, into the prior distributions.

1400 Cranfield Documents	
700 Test Documents	**700 Control Documents**
Rank documents for 100 *test queries* based on prior distributions and get relevance feedback data.	(1) Rank documents for 125 *training queries* based on prior distributions. (2) Rank documents for 125 *training queries* based on distributions updated using relevance data from training documents.

Table 1: *Evaluation method.*

Experiment 1 In the first set of experiments, we used only the past similar queries to specify the parameters $\{\alpha_{R_i}, \beta_{R_i}, \alpha_{\bar{R}_i}, \beta_{\bar{R}_i}\}$. We obtained the best retrieval performance using the similarity measure in Equation 5 with the threshold set to zero and using a weighted combination of the data from the past queries (Equation 8). With a threshold of zero, the new queries had an average of 33.7 similar past queries and every query had at least one similar query. At other thresholds, if a new query had no similar past queries, we set $\alpha_{R_i} = \beta_{R_i} = \alpha_{\bar{R}_i} = \beta_{\bar{R}_i} = 0.5$

Relevance feedback was most effective when we revised the variance of each Beta distribution such that the parameters α and β sum to one. We experimented with several variances such that the sum of α and β ranged from 1 to 100.

The precision-recall curve based only on these prior distributions is the solid line in Figures 1 and 2. All of the curves in these figures are for the "control group" of documents.

We also explored the effects of relevance feedback on retrieval performance. We were surprised to find that updating the prior distributions based on relevance data for the top-ranked documents actually *decreases* performance (Figure 1). We found that retrieving the top-ranked documents produced estimates of $P_q(t_i|\bar{R})$, for terms in the query, that are too large. This is due

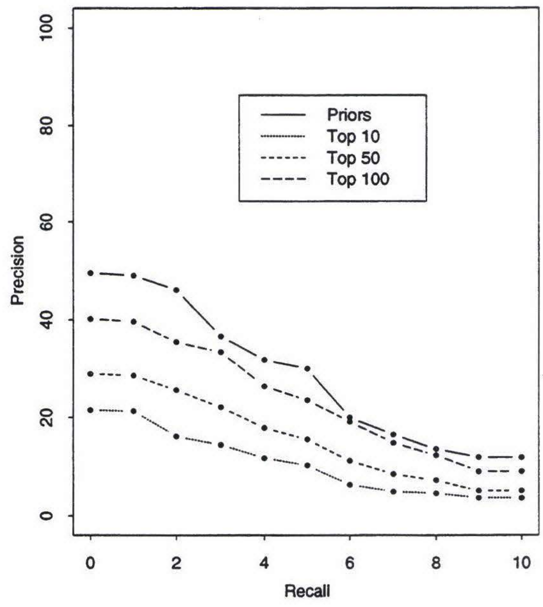

Figure 1: *Precision-recall curves based on distributions updated with the top-ranked documents.*

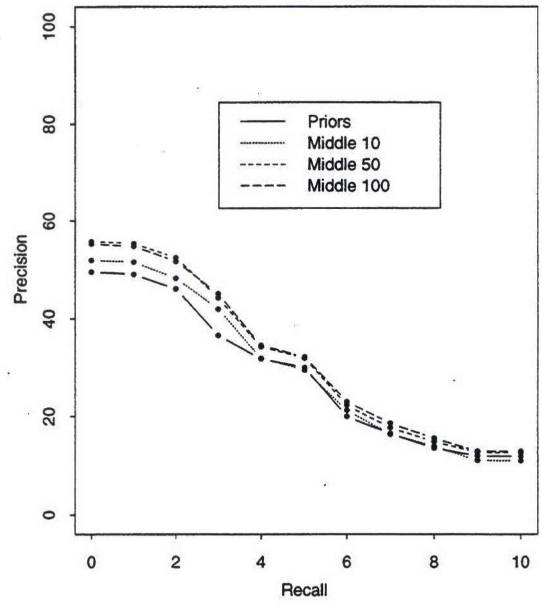

Figure 2: *Precision-recall curves based on distributions updated with the middle-ranked documents.*

to the fact that the non-relevant documents that are high in the ranking tend to have the query terms in larger proportions than in the entire set of non-relevant documents.

Lewis and Gale (1994) propose *uncertainty sampling* to train a classifier. They show that by requesting that the user label those documents for which we are least certain of their classification (here, relevance or non-relevance), we can increase effectiveness with fewer labeled documents. We also found this to be the case. Figure 2 shows that updating the prior distributions based on relevance judgments for the documents in the middle of the ranking improves retrieval performance.

Experiment 2 In a second set of experiments, we incorporated prior knowledge about query terms into the prior distributions. We base our knowledge on the results of experiments presented by Croft and Harper (1979) (see Section 5.2). Because they found that setting $P_q(t_i|R) = 0.9$ produced the best results, before looking at the data for past similar queries, we tried setting $\alpha_{R_i} = 9$ and $\beta_{R_i} = 1$. We then incorporated the past query information as in Experiment 1. Table 2 shows that specifying the prior distribution in this way (BIR+CH) produces better performance than the specification in Experiment 1 (BIR) and slightly better than Croft and Harper's method (CH).

Although increasing $P_q(t_i|R)$ in this way improves initial performance, when we update the distributions performance decreases (though only slightly). We found this to be the case regardless of how many or which documents we chose to retrieve. We are not sure why this is the case.

8 Future Work

The model that we have implemented makes two simplifying assumptions, that terms occurrences are binary and that terms are conditionally independent. However, the model is flexible enough that we may investigate relaxing these assumptions.

Though the assumption that the index terms are conditionally independent may not seem realistic; it remains to be seen what the implications are in practice. Langley et al. (1992) show that the "naive" Bayes approach is surprisingly effective in some applications. We may also explore alternative Bayesian classifiers such as Bayesian networks that account for inter-term dependencies.

The model also allows us to considered a non-binary term representation. For example, we could assume that the terms in relevant (and non-relevant) documents each come from a Poisson distribution. We would then place prior distributions on the expected frequency of occurrence of each term in the sets of relevant and non-relevant documents.

Recall %	Precision %		
	BIR	CH	BIR+CH
0	57.7	76.9	77.8
10	55.6	73.9	74.9
20	46.8	61.6	61.6
30	34.1	47.2	47.5
40	29.4	37.7	38.6
50	24.8	32.0	32.9
60	20.8	27.1	28.8
70	15.6	18.2	19.2
80	12.9	11.5	12.9
90	10.2	9.0	10.0
100	8.8	7.3	8.5

Table 2: *Precision-recall pairs comparing different specifications of the prior distributions.*

Another extension that we are considering is formalizing the approach of incorporating knowledge about queries into the prior distribution via a hierarchical Bayesian model.

Lastly, we would like to experiment on a larger test collection. We plan to use the OHSUMED test collection (Hersh et al. 1994) for our experiments. The OHSUMED test collection consists of Medline records from the years 1987 to 1991 which the National Library of Medicine has categorized by the Medical Subject Headings (MeSH). There are 233,445 records in the OHSUMED collection, each consisting of a title and abstract. We will treat the MeSH categories as text retrieval queries. Thus, the set of relevant documents are those that have been assigned to a particular MeSH category. This provides us with a large set of judged documents for use in training.

References

Bookstein, A. (1983). Information Retrieval: A Sequential Learning Process. *Journal of the American society for Information Science*, 34(5):331–342.

Croft, W. and Harper, D. (1979). Using Probabilistic Models of Document Retrieval Without Relevance Information. *Journal of Documentation*, 35(4):285–295.

Croft, W. B. and van Rijsbergen, C., editors (1994). *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Fox, E. A., Ingwersen, P., and Fidel, R., editors (1995). *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Fuhr, N. (1992). Probabilistic Models in Information Retrieval. *The Computer Journal*, 35(3):243–254.

Gelman, A., Carlin, J. B., Sterm, H. S., and Rubin, D. B. (1995). *Bayesian Data Analysis.* Chapman and Hall.

Hersh, W., Buckley, C., Leone, T., and Hickman, D. (1994). OHSUMED: an interactive retrieval evaluation and new large test collection for research. In Croft and van Rijsbergen (1994).

Langley, P., Iba, W., and Thompson, K. (1992). An analysis of Bayesian Classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 223–228. AAAI Press and M.I.T. Press.

Lewis, D. D. and Gale, W. (1994). A Sequential Algorithm for Training Text Classifiers. In Croft and van Rijsbergen (1994), pages 3–12.

Losee, R. (1988). Parameter Estimation for Probabilistic document-retrieval models. *Journal of the American Society for Information Science*, 39(1):8–16.

Maron, M. and Kuhns, J. (1960). On Relevance, Probabilistic Indexing and Information Retrieval. *Journal of the Association for Computing Machinery*, 7(3):216–244.

Raghavan, V. V. and Sever, H. (1995). On the Reuse of Past Optimal Queries. In Fox et al. (1995), pages 344–350.

Robertson, S. and Sparck Jones, K. (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3):129–146.

Salton, G., editor (1971). *The Smart Retrieval System–Experiments in Automatic Document Procesing.* Prentice-Hall, Englewood Cliffs, NJ.

Sparck Jones, K. and van Rijsbergen, C. (1976). Information Retrieval Test Collections. *Journal of Documentation*, 32(1):59–75.

van Rijsbergen, C. (1979). *Information Retrieval.* Butterworths, London, second edition.

Voorhees, E. M., Gupta, N. K., and Johnson-Laird, B. (1995). Learning Collection Fusion Strategies.