Model-based Feedback in the Language Modeling Approach to Information Retrieval

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

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

ABSTRACT

The language modeling approach to retrieval has been shown to perform well empirically. One advantage of this new approach is its statistical foundations. However, feedback, as one important component in a retrieval system, has only been dealt with heuristically in this new retrieval approach: the original query is usually literally expanded by adding additional terms to it. Such *expansion-based* feedback creates an inconsistent interpretation of the original and the expanded query. In this paper, we present a more principled approach to feedback in the language modeling approach. Specifically, we treat feedback as updating the query language model based on the extra evidence carried by the feedback documents. Such a *model-based* feedback strategy easily fits into an extension of the language modeling approach. We propose and evaluate two different approaches to updating a query language model based on feedback documents, one based on a generative probabilistic model of feedback documents and one based on minimization of the KL-divergence over feedback documents. Experiment results show that both approaches are effective and outperform the Rocchio feedback approach.

1. INTRODUCTION

The language modeling approach to text retrieval was first introduced by Ponte and Croft in [11] and later explored in [8, 5, 1, 15]. 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, make it an attractive framework in which to develop new text retrieval methodology.

Although the language modeling approach has performed well empirically, a significant amount of performance increase is often due to feedback [11, 8, 9]. Unfortunately, feedback has so far only been dealt with heuristically within the language modeling approach. In most existing work, it has been incorporated in an unnatural way: by expanding a query with a set of terms. But such an expansion-based feedback strategy is generally not very compatible with the essence of the language modeling approach, which is model estimation. As a result, the expanded query usually has to be interpreted differently than the original query. This is in contrast to the natural way of performing feedback in the classical relevance-based probabilistic model, such as the binary independence model [12].

In this paper, we propose a model-based approach to feedback that can be incorporated into the KL-divergence retrieval framework introduced in [6]. The model-based approach to feedback is actually not new; indeed, it is the essence of the classical probabilistic model [12]. However, it has been unclear how to incorporate model-based methods into the query-likelihood ranking function used in most existing work on the language modeling approach. We propose two different schemes for reestimating the query model based on a set of feedback documents:

1. *A generative model.* Assuming a generative model, we estimate the query topic model using the observed feedback documents based upon a maximum likelihood or regularized maximum likelihood criterion. The particular generative model we consider here is a simple mixture model, using the collection language model as one component, and the query topic model as the other.

2. *Divergence/risk minimization over feedback documents.* Here, rather than maximizing likelihood we estimate the query model by minimizing the average KL-divergence between the model and the feedback documents.

In the following section we provide a more detailed account of feedback techniques that have been used in previous work. Section 3 then introduces the KL-divergence framework for text retrieval, and Sections 4 and 5 present the new model-based frameworks for incorporating feedback. Section 6 presents the results of experiments carried out to evaluate these methods.

2. PREVIOUS FEEDBACK METHODS IN THE LM FRAMEWORK

Several recent papers have presented techniques for improving language modeling techniques using relevance or pseudo-relevance feedback. A ratio approach that selects terms having high probability in the feedback documents, but low probability according to the collection language model was proposed in [10]. The approach performs similarly to Rocchio [14] when very few relevant documents are used, but is significantly better than Rocchio when using more relevant documents. The pseudo relevance feedback results are also very promising, and significantly better than the results of using the baseline language modeling approach [10]. However, the ratio approach is conceptually restricted to the view of a query as a set of terms, and so cannot be naturally applied to the more general case when the query is considered as a *sequence* of terms and the frequency information of a query term is considered. Also, the number of terms needs to be determined heuristically.

Miller et al. [8] treat feedback as essentially expanding the original query with all terms in the feedback documents. Terms are pooled into bins by the number of feedback documents in which they occur, and for each bin, a different transition probability in the HMM is heuristically estimated. As a result, the smoothing is no longer equivalent to the simple linear interpolation, as it is in their basic HMM for smoothing the document language model. Thus, the model form *changes* as a result of incorporating feedback. Again, the interpretation of a query both as text (generated by an HMM) and as a set of terms is conceptually inconsistent. It also involves heuristic adjustment of transition probabilities by incorporating document frequency to filter out the high frequency words.

In [9], an approach is developed that is based on document likelihood ratios, and two interesting ideas concerning feedback are explored. First, a feedback criterion based on the optimization of the scores of feedback documents is developed, which turns out to be actually very similar to the ratio approach used in [11]. Second, a threshold for the number of selected terms is derived from the score optimization criterion. This approach is also reported to be effective [9], but shares the problem of inconsistent interpretation already mentioned. Other related work is [4], in which feedback documents are used to reestimate the smoothing parameters in the query-likelihood retrieval function. In effect, this is similar to query term reweighting in a traditional retrieval model, and does not fully take advantage of the feedback documents (e.g., no new terms are introduced to enhance a query).

Recent work has begun to develop model-based approaches to feedback, which appears to be a promising area for further development. In [6], an approach to feedback is developed that uses Markov chains to estimate a query model. While it is presented as a translation model [1], the Markov chain query expansion method, when applied to a set of feedback documents, can be regarded as a model-based approach as it reestimates the query language model. The relevance model estimation method proposed in [7] can also be used to estimate a richer query model based on feedback documents. Both approaches rely on the query words to focus the model. In the methods proposed here, we work with the feedback documents alone, and estimate a query model that can be used to update an existing query model.

3. THE KL-DIVERGENCE RETRIEVAL MODEL

In general, any approach to the retrieval problem is decomposed into three basic components: (1) query representation; (2) document representation; and (3) matching of query representation and document representation. In the KL-divergence model, these components are realized in the following probabilistic way. First, we assume that a query (or document) can be viewed as an observation from a probabilistic query (or document) model. The representation problem is thus equivalent to that of model estimation. Second, the relevance value of a document with respect to a query is measured by the Kullback-Leibler divergence between the query model and document model. The matching problem is thus equivalent to measuring the similarity or "distance" between the estimated query model and document model. The KL-divergence retrieval model was introduced in [6] as a special case of the more general risk minimization retrieval framework. Interestingly, it is similar to the vector space model, except that we use language models, rather than ordinary term vectors to represent a document or a query.

We now present the model more formally. Given two probability mass functions $p(x)$ and $q(x)$the Kullback-Leibler divergence (or relative entropy) between p and q, denoted $D(p \,\|\, q)$, is defined as

$$D(p \,\|\, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

It is easy to show that $D(p \,\|\, q)$ is always non-negative and is zero if and only if $p = q$. Even though it is not a true distance between distributions (because it is not symmetric and does not satisfy the triangle inequality), it is still often useful to think of the KL-divergence as a "distance" between distributions [2].

Now, assume that a query $\mathbf{q}$ is obtained as a sample from a generative model $p(\mathbf{q} \,|\, \theta_Q)$ with parameters θ_Q. Similarly, assume that a document $\mathbf{d}$ is generated by a model $p(\mathbf{d} \,|\, \theta_D)$ with parameters θ_D. If $\widehat{\theta}_Q$ and $\widehat{\theta}_D$ are the estimated query and document language models respectively, then, according to [6], the relevance value of $\mathbf{d}$ with respect to $\mathbf{q}$ can be measured by the following KL-divergence function:

$$D(\widehat{\theta}_Q \,\|\, \widehat{\theta}_D) \quad = \quad -\sum_w p(w \,|\, \widehat{\theta}_Q) \log p(w \,|\, \widehat{\theta}_D) + cons(\mathbf{q})$$

The document-independent constant $cons(\mathbf{q})$ (the entropy of the query model) can be dropped, because it does not affect ranking of documents, so ranking based on the risk is equivalent to ranking based on the cross entropy of the query language model with respect to the document language model. The minimum value (i.e., query model entropy) is achieved when $\widehat{\theta}_D$ is identical to $\widehat{\theta}_Q$, which makes perfect sense for retrieval. The popular query-likelihood ranking function, used in most of the previous work on the language modeling approach, is easily obtained as a special case of the KL-divergence model when the query model is estimated as the empirical distribution of the query.

Although the KL divergence model appears to be similar to the probability distribution model proposed in [17] (when the information-theoretic retrieval strategy is used), it is actually much more general and flexible because of its explicit modeling of the query and documents. In [17], the multinomial term distribution is proposed as primarily an *alternative representation* of documents and query (in the sense of the vector-space model), not a generative model for documents or query. Thus, it is not surprising that the issue of model estimation has not been considered at all and the term distribution representation is naturally assumed to be best approximated by the relative frequency of terms. Thus, model smoothing has not been considered as a possibility in this work.

Within the KL-divergence model, the retrieval problem is essentially equivalent to the problem of estimating $\widehat{\theta}_Q$ and $\widehat{\theta}_D$. In principle, we can use any language model for the query and document. Such flexibility makes the model quite general and allows us to model a query or document in different ways. For example, if a collection is regarded as a "document," then the model can be used for distributed information retrieval. Interesting work in this direction by Xu and Croft [18] estimates a topic model based on a set of example documents and then uses the KL-divergence to select topic models for a query.

Our approach relies on the estimation of both document and query language models. The lack of a query model in previous work on the language modeling approach has made it unnatural to incorporate feedback, a very important retrieval technique. We view the introduction of a query language model as a necessary step toward more powerful retrieval methods based on language modeling. We assume that the user's topic (information need) may be modeled/represented by a language model, in the simplest case a unigram model. As the model is expected to generate text indicating the user's information need, our task is to estimate the underlying model by exploiting *all* the information we know about that information need. In the traditional setup there are two major pieces of information from the user that may help us infer the model: the query and the judged relevant documents. In this paper, we explore simple smoothing strategies for combining the relevant set with the query; the simplest is based on linear interpolation. Specifically, let $\widehat{\theta}_Q$ be the original query model and let $\widehat{\theta}_{\mathcal{F}}$ be an estimated feedback query model based on feedback documents $\mathcal{F} = (d_1, d_2, ..., d_n)$, which can be the documents judged to be relevant by a user, or the top documents from an initial retrieval (as in the case of pseudo relevance feedback). Then, our new query model $\widehat{\theta}_{Q'}$ is

$$\widehat{\theta}_{Q'} = (1-\alpha)\,\widehat{\theta}_Q + \alpha\,\widehat{\theta}_{\mathcal{F}}$$

where α controls the influence of the feedback model. In the following sections, we describe two very different strategies for estimating $\widehat{\theta}_{\mathcal{F}}$ based on feedback documents.

4. A GENERATIVE MODEL OF FEEDBACK DOCUMENTS

A natural way to estimate a feedback query model $\widehat{\theta}_{\mathcal{F}}$ is to assume that the feedback documents are generated by a probabilistic model $p(\mathcal{F} \mid \theta)$. One of the simplest generative models is a unigram language model, which generates each word in $\mathcal{F}$ independently according to θ. That is,

$$p(\mathcal{F} \mid \theta) = \prod_i \prod_w p(w \mid \theta)^{c(w;d_i)}$$

where $c(w; d_i)$ is the count of word w in document d_i. This simple model would be reasonable if our feedback documents only contain relevant information. However, most documents probably also contain background information or even non-relevant topics. A more reasonable model would be a mixture model that generates a feedback document by mixing the query topic model with a collection language model. That is, a document is generated by picking a word using either the query topic model $p(w \mid \theta)$ or the collection language model $p(w \mid \mathcal{C})$. The collection language model is a reasonable model of the irrelevant content in a feedback document.

Under this simple mixture model, the log-likelihood of feedback documents is

$$\log p(\mathcal{F} \mid \theta) = \sum_i \sum_w c(w; d_i) \log((1-\lambda) p(w \mid \theta) + \lambda\, p(w \mid \mathcal{C}))$$

Note that if both λ and θ are to be estimated, then the maximum likelihood estimate of λ would be zero and our mixture model would reduce to a simple unigram model. Intuitively, however, we should like to have a non-zero λ, indicating the amount of background "noise" when generating a document. Thus, we will set λ to some constant and estimate only θ, which can be done by using the EM algorithm [3]. The EM updates for $p_\lambda(w \mid \widehat{\theta}_{\mathcal{F}})$ are:

$$t^{(n)}(w) = \frac{(1-\lambda) p_\lambda^{(n)}(w \mid \theta_{\mathcal{F}})}{(1-\lambda) p_\lambda^{(n)}(w \mid \theta_{\mathcal{F}}) + \lambda p(w \mid \mathcal{C})}$$

$$p_\lambda^{(n+1)}(w \mid \theta_{\mathcal{F}}) = \frac{\sum_{j=1}^n c(w; \mathbf{d}_j) t^{(n)}(w)}{\sum_i \sum_{j=1}^n c(w_i; \mathbf{d}_j) t^{(n)}(w_i)}$$

Intuitively, when estimating the query model, we are trying to "purify" the document by eliminating some background noise. Thus, the estimated query model will generally be concentrated on words that are common in the feedback document set, but not very common according to the collection language model $p(\cdot \mid \mathcal{C})$. This is precisely the effect that most traditional feedback methods, such as Rocchio [14], try to capture.

To score a document $\mathbf{d}$ using the estimated query model $\widehat{\theta}_{\mathcal{F}}$, we first interpolate it with the original query model $\widehat{\theta}_Q$ to obtain an updated query model $\widehat{\theta}_{Q'}$, and then compute the KL-divergence between $p(\cdot \mid \widehat{\theta}_{Q'})$ and $p(\cdot \mid \widehat{\theta}_D)$, where $\widehat{\theta}_D$ is the smoothed empirical word distribution of $\mathbf{d}$.

5. DIVERGENCE MINIMIZATION OVER FEEDBACK DOCUMENTS

A different strategy for estimating a query model based on feedback documents is to minimize the divergence between the model and the feedback documents. Let $\mathcal{F} = (\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_n)$ be a set of feedback documents. We define the empirical KL-divergence between the query model θ over $\mathcal{F}$ and the feedback documents as

$$D_e(\theta;\mathcal{F}) \quad = \quad \frac{1}{|\mathcal{F}|}\sum_{i=1}^{n} D(\theta \,\|\, \widehat{\theta}_{\mathbf{d}_i})$$

That is, as the average divergence between the smoothed empirical word distribution of each document ($\widehat{\theta}_{\mathbf{d}_i}$).

Intuitively, if we estimate the query model by minimizing this average divergence, we will have a query model that, when used to score documents, will give us the best average score over the feedback documents. The estimated query model will be close to each feedback document model; however, since feedback documents typically share many common words due to the language and domain characteristics, such a query model may be quite general. One way of specializing the model is to add a regularization term to the divergence function. We do this by preferring a model that incurs a *greater* divergence with respect to the collection model, which is an approximation of the language model for off-topic or background content.

Incorporating this condition, we end up with the following empirical divergence function of a feedback query model:

$$D_e(\theta;\mathcal{F},\mathcal{C}) \quad = \quad \frac{1}{|\mathcal{F}|}\sum_{i=1}^{n} D(\theta \,\|\, \widehat{\theta}_{\mathbf{d}_i}) - \lambda D(\theta \,\|\, p(.\,|\,\mathcal{C}))$$

Here $\lambda \in [0,1)$ is a weighting parameter, and $p(w\,|\,\mathcal{C})$ is the collection language model. Minimizing this divergence is equivalent to maximizing the entropy of the model under a preference constraint encoded in the second term. This is very similar to the maximum entropy approach to parameter estimation. Using this criterion, our estimate $\widehat{\theta}_{\mathcal{F}} = \arg\min_{\theta} D_e(\theta;\mathcal{F},\mathcal{C})$ is then given by

$$p(w\,|\,\widehat{\theta}_{\mathcal{F}}) \;\propto\; \exp\left(\frac{1}{(1-\lambda)}\frac{1}{|\mathcal{F}|}\sum_{i}\log p(w\,|\,\widehat{\theta}_{d_i}) - \frac{\lambda}{1-\lambda}\log p(w\,|\,\mathcal{C})\right)$$

We see that the resulting model assigns a high probability to words that are common in the feedback documents, but not common according to the collection language model. The parameter λ controls the weight on the collection language model. Similar to the λ in the collection mixture model, when λ is set to zero, the effect of the collection language model is completely ignored, and we then have a query model that strictly minimizes the divergence over the feedback documents. In this case the model is given by the geometric mean of the distributions of the feedback documents.

As before, to exploit $\widehat{\theta}_{\mathcal{F}}$ in our KL-divergence retrieval model, we first interpolate it with the original query model $\widehat{\theta}_Q$ to obtain an updated model $\widehat{\theta}_{Q'}$, and then score a document $\mathbf{d}$ by $D(\widehat{\theta}_{Q'} \,\|\, \widehat{\theta}_{\mathbf{d}})$.

6. EXPERIMENTS

The KL-divergence retrieval framework allows us to combine any pair of document and query language models; thus, experimentally there can be many possible combinations to explore. In this paper, we fix the document language model and focus on different ways of estimating the query model based on feedback documents. Specifically, we use a Dirichlet prior (with a hyperparameter of 1,000) for estimating the document language models in all the experiments. In effect, this interpolates the maximum likelihood estimate of the document language model with the collection language model using a document-dependent interpolation coefficient of $1000/(1000+|d|)$ for the collection model. This approach is described in detail and evaluated experimentally in [19]. An appropriate way of evaluating a feedback method would be to consider both relevance feedback and pseudo (or blind) feedback, but as a first step, we only consider pseudo feedback in this paper. In all experiments, we take the top 10 documents from a set of previously retrieved results obtained using the basic query-likelihood ranking function and Dirichlet smoothing. We compare the query models estimated using the collection mixture and the divergence minimization methods described in the previous sections, varying both the interpolation parameter (α) and the feedback model estimation parameters (λ).

6.1 Testing Collections and Evaluation

We evaluated both feedback approaches on three TREC collections [16]:

1. *AP88&89 with topics 101-150.* This is the same as one of the collections used in [7], and will be labeled as "AP88-89".
2. *TREC Disk 4&5 (minus Congressional Record) with topics 401-450.* This is the official TREC8 ad hoc task collection, and will be labelled as "TREC8".
3. *TREC8 small web collection with topics 401-450.* This is the official TREC8 small web task collection, and will be labelled as "WEB".

In all cases, we use only the titles of the topic description, since they are closer to the actual queries used in real applications, and since feedback is expected to be most useful for short queries. We have done minimal preprocessing of documents and queries; the only tokenization performed is stemming (using a Porter stemmer), and no stopword list is applied. We believe that with appropriate probabilistic modeling, stop words can be effectively down-weighted. In each run, the top 1,000 documents are returned and evaluated, as is commonly done in TREC evaluations.

The following performance measures are considered in our evaluation:

- Interpolated precision at different, but fixed, recall levels (i.e., the PR curve)
- Initial precision; that is, the best precision achievable at any document cutoff
- Non-interpolated average precision
- Recall at 1,000 documents

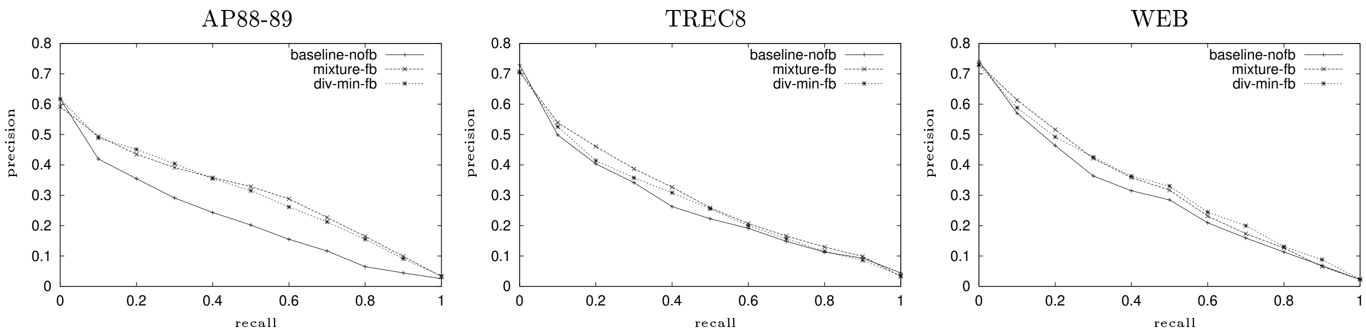

Figure 1: Effect of feedback on AP88-89 (left), TREC8 (middle), and WEB (right). In each plot, the two feedback methods are compared with the baseline simple language modeling approach (no feedback).

Collection		Simple LM	Mixture FB	Improv.	Div. Min.	Improv.
AP88-89	AvgPr	0.210	0.296	+41%	0.295	+40%
	InitPr	0.617	0.591	−4%	0.617	+0%
	Recall	3067/4805	3888/4805	+27%	3665/4805	+19%
TREC8	AvgPr	0.256	0.282	+10%	0.269	+5%
	InitPr	0.729	0.707	−3%	0.705	−3%
	Recall	2853/4728	3160/4728	+11%	3129/4728	+10%
WEB	AvgPr	0.281	0.306	+9%	0.312	+11%
	InitPr	0.742	0.732	−1%	0.728	−2%
	Recall	1755/2279	1758/2279	+0%	1798/2279	+2%

Table 1: Comparison of the basic language modeling method with model-based feedback methods. Column three and five give the performance using the mixture model and divergence minimization respectively.

The performance over a query set is reported as the average of the corresponding performance figures for individual queries (i.e., the so-called "macro" average), except that the average recall is actually the total number of retrieved relevant documents for all queries divided by the total count of relevant documents (i.e., the so-called "micro" average). We take the average precision as the primary single summary performance for an experiment, as it reflects the overall ranking accuracy well, though we sometimes also report other measures.

6.2 The Effect of Feedback

In order to see the effect of feedback, we compare the feedback results with the baseline non-feedback results. In general, we find that, with appropriate parameter settings, both feedback techniques that we propose can be very effective. For example, the best feedback results from each method are compared with the baseline performance in Figure 1 and Table 1. The average precision and recall are consistently improved by performing feedback. The increase in average precision is larger than 10% in most cases. We also note that the initial precision of feedback results is slightly decreased in almost all cases. Given that not all of the top ten documents may be relevant, this is not very surprising, as the initial precision is very sensitive to the ranking of one particular document on the top, while our goal is to improve the overall ranking of documents. It is interesting that the improvement on AP88-89 is much greater than that on TREC8 and WEB. This seems to be true for both approaches and also true for the Rocchio approach to be discussed below, suggesting that feedback on AP88-89 is somehow "easier" than on TREC8 or WEB (e.g., because of the homogeneity of documents). Further experiments and analysis are needed to understand this better.

In Table 2, we compare our feedback results with that of a tuned Rocchio approach with TF-IDF weighting. The TF formula used is the one based on the BM25 retrieval formula with the same parameter settings as presented in [13]. We fixed the number of documents for feedback (top 10), and varied the two main parameters in Rocchio—the coefficient and the number of terms. The reported results are the best results we obtained. Note that these Rocchio baseline results are actually very strong when compared with the published official TREC8 and WEB results, especially when considering that we used only title queries [16]. When compared with the Rocchio results, the two model-based feedback methods both perform better in terms of precision, though their recall is often slightly worse than Rocchio.

We suspect that the decrease in recall may be because we tuned the number of terms to use in the Rocchio method, but have not tuned the probability cutoff used in our methods, which essentially controls the number of terms to introduce for feedback. Indeed, in all of the experiments, we

Collection		Rocchio FB	Mixture FB	Improv.	Div. Min. FB	Improv.
AP88-89	AvgPr	0.291	0.296	+2%	0.295	+1%
	InitPr	0.566	0.591	+4%	0.617	+9%
	Recall	3729/4805	3888/4805	+4%	3665/4805	−3%
TREC8	AvgPr	0.260	0.282	+8%	0.269	+3%
	InitPr	0.657	0.707	+8%	0.705	+7%
	Recall	3204/4728	3160/4728	−1%	3129/4728	−2%
WEB	AvgPr	0.271	0.306	+13%	0.312	+15%
	InitPr	0.600	0.732	+22%	0.728	+21%
	Recall	1826/2279	1758/2279	−4%	1798/2279	−2%

Table 2: Comparison of the Rocchio feedback method with model-based feedback methods. Column three and five give the performance of using the mixture model and divergence minimization respectively.

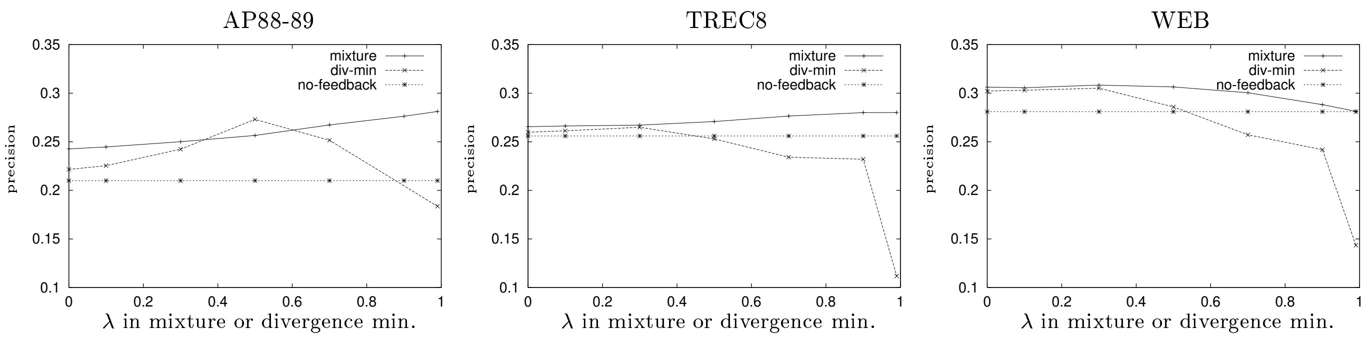

Figure 2: Sensitivity of precision to feedback model parameters on AP88-89 (left), TREC8 (middle), and WEB (right). In each plot, the horizontal line is the non-feedback performance, and the other two lines correspond to the two feedback methods respectively. Note that the x-axis means different λ for different methods. For each dataset, the interpolation coefficient was set to $\alpha = 0.5$.

truncated the estimated query model by ignoring all terms having a probability less than 0.001. It is reasonable to expect the recall to be improved when using a lower probability cutoff. Note that the precision can be expected to stay the same or increase as well when more terms are selected, because the extra terms generally have a very small probability, and so will be unlikely to have a great impact on the ranking of documents with high scores.

The comparisons made here are all based on some of the best feedback results. It is therefore important that we also study how feedback performance may be affected by the choice of parameters in our model. We first look at the sensitivity to the parameter in each feedback method.

6.3 Sensitivity of Performance to Feedback Model Parameter

In the mixture model method, the parameter λ controls the amount of "background noise" in the feedback documents, while in the divergence minimization method, the parameter λ controls the influence of the collection language model, which is included in a geometric mean. In both cases, λ indicates the extent to which the estimated query model should be deviate from the collection language model. Although the two λ's play a similar role conceptually, we find that they affect the feedback performance in very different ways. This difference can be seen in Figure 2, in which we show how the average precision changes according to different values of λ, for the fixed value $\alpha = 0.5$. Specifically, we see that the performance is relatively insensitive to the setting of λ in the mixture model method, but can be quite sensitive to the setting of λ in the divergence minimization method. Indeed, with $\alpha = 0.5$, the mixture model performance is generally above the baseline, no matter which value we set λ to. However, the divergence minimization performance is only above the baseline when λ is small. When λ is large, the performance is extremely bad and significantly worse than the baseline performance.

6.4 Influence of the interpolation coefficient

Recall that we interpolate the estimated feedback query model with the original maximum likelihood model estimated based on the query text. The interpolation is controlled by a coefficient α. When $\alpha = 0$, we are only using the original model (i.e., no feedback), while if $\alpha = 1$, we completely ignore the original model and use only the estimated feedback model. In the actual experiments, we truncated the estimated feedback model by ignoring all terms with a probability lower than 0.001, and renormalized it before interpolating.

Figure 3 shows how the average precision under feedback varies according to the value of α. Each line represents a

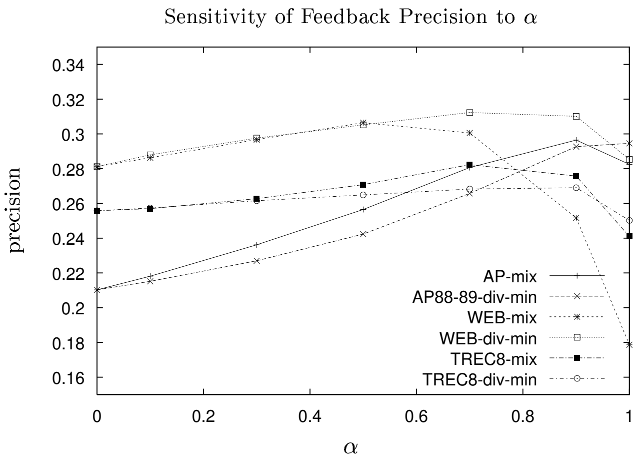

Figure 3: Influence of α value on precision. Lines represent different feedback models on different testing collections.

specific feedback model (estimated using either the mixture model or the divergence minimization method) on a particular test collection. Note that the precision at $\alpha = 0$ is actually the baseline non-feedback performance and the precision at $\alpha = 1$ is the performance resulting from using only the feedback model.

We see that the setting of α can affect the performance significantly. For example, on AP88-89, the feedback model alone is much better than the original query model, thus the optimal setting of α tends to be close to 1. On the other hand, on both TREC8 and WEB, the feedback model alone is much worse than that of the original query model, but when it is interpolated with the original query model appropriately, it can be much more effective than either model alone. This means that the two models complement each other well. The original query model helps focus on the topic, while the feedback model supplements it by suggesting related words. The precision of the mixture model method appears to be more sensitive to α than the precision of the divergence minimization method is, especially on the WEB collection. It appears that it is usually safe to set α to a value close to, but smaller than 0.5.

7. CONCLUSIONS

In this paper, we propose two *model-based* methods for performing feedback in the language modeling approach to information retrieval. This is in contrast to the *expansion-based* feedback methods used in most existing work. One advantage of the model-based approach is that it maintains conceptual consistency when interpreting the query in the retrieval model, and it explicitly treats the use of feedback as a learning process.

In both methods proposed, the feedback documents are used to estimate a query model, which is then used to update the original query model with linear interpolation. The two methods differ in the way they estimate the query model based on the feedback documents. The first method assumes the feedback documents are generated by a mixture model in which one component is the query topic model and the other is the collection language model. Given the observed feedback documents, the maximum likelihood criterion is used to estimate a query topic model. The second method uses a completely different estimation criterion, chosing the query model that has the smallest average KL-divergence from the smoothed empirical word distribution of the feedback documents.

The two methods were evaluated on three representative large retrieval collections. The results show that both methods are effective for feedback and perform better than the Rocchio method in terms of precision. Analysis of the results indicates that the performance can be sensitive to the settings of the interpolation coefficient α as well as to the parameter λ in each feedback method. The precision of the mixture model tends to be more sensitive to α than that of the divergence minimization method. On the other hand, the precision is relatively insensitive to λ in the mixture model method, but it is very sensitive to λ in the divergence minimization method. It appears that setting α to a value close to, but smaller than, 0.5, is good in most cases. A smaller λ (e.g., $\lambda = 0.3$) is probably appropriate for divergence minimization; while the λ in the mixture model method can be set to 0.5.

Although these patterns are observed on feedback with only 10 documents, in other experiments that have not been reported here we found that with more feedback documents (e.g., 50), the sensitivity pattern appears to be basically the same as what we reported here, and the performance gain from feedback is usually even more. Obviously, as we use more and more documents, the performance will eventually decrease. The fact that we have very little control over the true relevant examples is a serious drawback in experimenting with pseudo feedback only; it is often hard to tell if inferior feedback performance is due to poor technique or just due to errors and noise in the feedback examples. An extreme case would be that the top 10 documents are all non-relevant because of a bad initial ranking. Obviously, we cannot expect any feedback technique to gain much in this case. Thus, an important consideration for future work is to test the proposed feedback techniques for *relevance* feedback, in which we will be able to examine the effectiveness of learning more closely. A related direction is to consider our confidence in assuming all of the top 10 documents to be relevant. We would like to associate a relevance probability with each feedback document, so that the estimated query model will be affected more by those documents having a higher relevance probability.

8. ACKNOWLEDGEMENTS

This research was sponsored in full by the Advanced Research and Development Activity in Information Technology (ARDA) under its Statistical Language Modeling for Information Retrieval Research Program.

9. 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] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38, 1977.

[4] D. Hiemstra. *Using language models for information retrieval*. PhD thesis, University of Twente, 2001.

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

[6] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR'2001*, Sept 2001.

[7] V. Lavrenko and B. Croft. Relevance-based language models. In *Proceedings of SIGIR'2001*, Sept 2001.

[8] 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.

[9] K. Ng. A maximum likelihood ratio information retrieval model. In *TREC-8 Workshop notebook*, 1999.

[10] J. Ponte. *A Language Modeling Approach to Information Retrieval*. PhD thesis, Univ. of Massachusetts at Amherst, 1998.

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

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

[13] S. E. Robertson and S. Walker. Okapi/keenbow at TREC-8. In E. M. Voorhees and D. K. Harman, editors, *The Eighth Text REtrieval Conference (TREC 8)*. NIST Special Publication 500-246, 1999.

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

[15] F. Song and B. Croft. A general language model for information retrieval. In *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–280, 1999.

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

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

[18] J. Xu and W. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the SIGIR 1999*, pages 254–261, 1999.

[19] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'2001*, Sept 2001.