A Mixture Clustering Model for Pseudo feedback in Information Retrieval

Tao Tao and ChengXiang Zhai

Department of Computer Science, University of Illinois at Urbana-Champaign

1 Introduction

Information Retrieval (IR) refers to retrieving relevant documents from a large document database according to a user-submitted query, and is among the most useful technologies to overcome information overload. For example, Web search engines are now essential tools for everyone to find information on the Web. Indeed, search capabilities are becoming more and more popular in virtually all kinds of information management applications.

Given a query, a retrieval system would typically estimate a relevance value for each document w.r.t. this query, and rank the documents in the descending order of relevance. Over the decades, many different retrieval models have been proposed and tested, including vector space models , probabilistic models , and logic-based models [6]. As a special family of probabilistic models, the language modeling approaches have attracted much attention recently due to their statistical foundation and empirical effectiveness [5, 2].

A particular effective retrieval model based on statistical language models is the Kullback-Leibler (KL) divergence unigram retrieval model proposed and studied in [4, 8]. The basic idea of this model is to measure the relevance value of a document w.r.t. a query by the Kullback-Leibler divergence between the corresponding query model and the document model. Thus the retrieval task essentially boils down to estimating a query unigram model[1] and a set of document unigram language models. The retrieval accuracy is largely affected by how good the estimated query and document models are. In this paper, we study how to improve the query model estimation through fitting a mixture model to some number of top ranked documents, which are retrieved by the original query itself. We present a new mixture model that extends and improves an existing mixture feedback model and addresses its two deficiencies. We study parameter estimation for this mixture model, and evaluate the model on a document set with $160,000$ news article documents and 50 queries. The results show that using the new mixture model not only

[1] A unigram language model is just a multinomial word distribution.

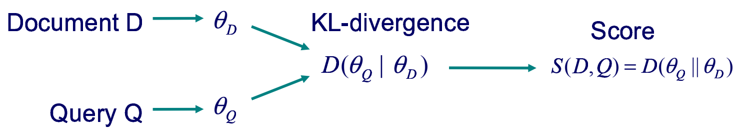
Document D
θ_D
KL-divergence
$D(\theta_Q \mid \theta_D)$
Score
$S(D,Q) = D(\theta_Q \| \theta_D)$
Query Q
θ_Q

Fig. 1. The KL-divergence Retrieval Model

significantly improves the retrieval performance over using the original query model, but also performs better than the old mixture model.

The rest of the paper is organized as follows: First, in Section 2, we provide some details about the KL-divergence retrieval formula as background for understanding the mixture problem estimation. We then present our mixture model and its estimation in Section 3. Finally, experiment results are presented in Section 4.

2 The Kullback-Leibler divergence retrieval model

The basic idea of the KL-divergence model is to score a document w.r.t. a query based on the KL-divergence between an estimated document model and an estimated query model. 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

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

Now, assume that a query $\mathbf{q}$ is obtained as a sample from a unigram language model (*i.e.*, a multinomial word distribution) $p(\mathbf{q} \,|\, \theta_Q)$ 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 the relevance value of $\mathbf{d}$ w.r.t. $\mathbf{q}$ can be measured by $D(\widehat{\theta}_Q||\widehat{\theta}_D)$.

The KL-divergence model contains three independent components: (1) the query model $\widehat{\theta}_Q$; (2) the document model $\widehat{\theta}_D$; and (3) the KL-divergence function. (See Fig. 1 for an illustration.). Given that we fix the KL-divergence function, the whole retrieval problem essentially boils down to the problem of accurately estimating a query model and a set of document models.

The simplest generative model of a document is just the unigram language model θ_D, a multinomial distribution. Usually a smoothing method is applied to avoid overfitting [9]. The simplest generative model for a query is also just a unigram language model, which can be estimated as the relative frequency of the words in the query. Generally, a query is too short to estimate a query model accurately. A general heuristic approach used in information retrieval is the so-called "pseudo feedback". The basic idea is to assume a small number of top-ranked documents from an initial retrieval result to be relevant, and use them to refine the query model. Presumably, a relevant document can provide a lot of information about what a user is interested in, thus can be expected to help improve the estimated query model. Even though not all the top-ranked documents are actually relevant, we can expect some of them to be relevant and those non-relevant ones are also similar to a relevant document. For this reason, pseudo feedback in general leads to improvement of average retrieval accuracy.

In [8], a general two-step pseudo feedback procedure is proposed, in which we first estimate a feedback topic model $\hat{\Theta}_F$ based on a set of feedback documents (e.g., top 5 documents) F, and then update the original query model $\hat{\Theta}_Q$ through heuristically interpolating $\hat{\Theta}_Q$ with $\hat{\Theta}_F$ to obtain a new query model $\theta_{Q'}$ i.e., $\hat{\Theta}_{Q'} = (1 - \alpha)\Theta_Q + \alpha\hat{\Theta}_F$, where $\alpha \in [0, 1]$ is a parameter to control the influence of feedback.

Two specific methods are proposed in [8] to estimate the feedback model Θ_F, one being based on a mixture model and one on divergence minimization. Although both methods have been shown to be quite effective, the separation of the original query model from the estimation of the feedback model makes it hard to automatically tune the feedback parameters. More specifically, the separation causes two problems: (1) It makes it hard to discriminate the feedback documents when estimating the feedback model. Presumably, we should trust the top ranked documents more than the lowly ranked ones. But without involving the original query model, it is difficult to implement this intuition in a principled way. Without implementing this intuition, the feedback performance will be very sensitive to the number of documents to use for pseudo feedback. (2) It is difficult to automatically tune the interpolation coefficient, since this parameter is now outside our feedback model. In this paper, we extend this work and develop a new mixture model that will incorporate the original query model as a prior when estimating the feedback model. In essence, we perform *biased clustering* of words in the feedback documents with one cluster "anchored" to our query model and the other to more general vocabulary. A key novel feature of the new mixture model is that it does *not* assume that all the feedback documents have the same amount of relevance information to contribute to the new query model, which presumably helps address the first problem. In addition, the incorporation of the original query model as a prior integrates the two steps and makes it possible to tune the feedback parameters automatically with the data. We thus expect the new mixture model to be more robust than the original mixture model proposed in [8].

3 A mixture clustering model for pseudo feedback

In this section, we present our new mixture model in detail. We first define the following notations. Q is a query. C is the set of all documents in the whole collection (*i.e.*, document database). $D = \{d_1, ..., d_k\}$ is a set of documents as feedback. We use w and d to represent an individual word and document and $c(w,d)$ $(c(w,Q))$ to mean the count of word w in document d (query Q).

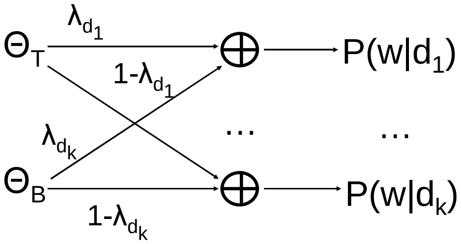

Fig. 2. Mixture model for pseudo feedback.

Our general idea is to regard the original (current) query model θ_Q as inducing a prior on the true query model θ_T, $p(\theta_T|\theta_Q)$, and view the feedback documents D as providing new evidence about the true query model. We then use Bayesian estimation to obtain a (presumably better) query model θ_T.

$$\hat{\theta}_T = \arg\max_{\theta_T} p(D|\theta_T)p(\theta_T|\theta_Q) \tag{1}$$

The feedback document sampling model is a mixture generative model for the feedback documents, where each document is assumed to be "generated" from a two-component unigram mixture model $P(w|d_i)$, one being the topic language model θ_T , which intends to capture the relevance information in the feedback documents, and one being a background language model θ_B capturing the general English usage and any distracting non-relevant information. Intuitively, we are assuming that a feedback document is "written" by sampling words in such a way that we sometimes draw words according to θ_T and sometimes according to θ_B. The *document-dependent* mixing weight parameter $\lambda_d \in [0,1]$ determines how often we would sample a word using the topic model θ_T. In effect, this model allows us to perform a biased clustering of the words in D where one cluster is anchored to the query model while the other to general background vocabuary. The model is illustrated in Fig. 2.

According to this mixture model, the log-likelihood for the feedback documents is

$$L(\Lambda|D) = \sum_{i=1}^{k} \sum_{w \in V} c(w,d_i) \log(\lambda_{d_i} p(w|\theta_T) + (1-\lambda_{d_i})p(w|\theta_B))$$

where $\Lambda = (\theta_T, \theta_B, \lambda_1, ..., \lambda_k)$ is all the parameters and V is the vocabulary.

To regulate the mixture model, we will fix the background model θ_B to some unigram language model estimated using all the documents in the collection C, since most of them are non-relevant. We thus only need to estimate θ_T and λ_{d_i}'s. θ_T is meant to be our new (presumably improved) query model, and λ_{d_i}'s are meant to model the amount of relevant information in each document and thus allow us to discount feedback documents appropriately. To incorporate the original query model, we use it to define a Dirichlet conjugate prior for θ_T, and estimate θ_B and λ_{d_i}'s using the Maximum A Posterior (MAP) estimator. We will also put a conjugate prior (a beta distribution) on the λ_{d_i}'s, which encodes our prior belief of the amount of relevant information in each feedback document.

The MAP estimate can be implemented using the standard EM algorithm with some slight modification to the M-step to incorporate the prior pseudo counts [3], leading to the following updating formulas:

$$Z_{w,d} = \frac{\lambda_d^{(n)} p^{(n)}(w|\theta_T)}{\lambda_d^{(n)} p^{(n)}(w|\theta_T) + (1-\lambda_d^{(n)})p(w|\theta_B)} \tag{2}$$

$$\lambda_d^{(n+1)} = \frac{\mu\lambda_{prior} + \sum_{w\in V} c(w,d)Z_{w,d}}{\mu + \sum_{w\in V} c(w,d)} \tag{3}$$

$$p^{(n+1)}(w|\theta_T) = \frac{\sigma k p(w|\theta_Q) + \sum_{d\in D} c(w,d)Z_{w,d}}{\sigma k + \sum_{w'\in V}\sum_{d\in D} c(w',d)Z_{w',d}} \tag{4}$$

where λ_{prior} is the mean of the beta prior for λ_d, $p(w|\theta_Q)$ is the original query model (defining the Dirichlet prior mean), and σ and μ are our confidence on λ_{prior} and the original query model prior (*i.e.*, prior equivalent sample size), respectively. Note that k is the number of feedback documents, and we parameterize the confidence on the query model prior with σk so that σ can be interpreted as the equivalent sample size relative to each document.

On the surface, it appears that we now have more parameters to set than in the old model. However, all the parameters can be set in a meaningful way, and with appropriate regulation, we can hope the model not to be sensitive to the setting of all parameters.

First, σ encodes our confidence on the original query model, corresponding to the expected amount of feedback. It can be interpreted as the "equivalent sample size" of our prior as compared with one *single* feedback document. Thus if σ is set to 10, the original query model would influence the estimated query model as much as a *completely* relevant document with $10 \times k$ words. This setting is found to be optimal in all our experiments.

Second, λ_{prior} corresponds to our prior of how much relevance information is in each document. A smaller λ_{prior} would cause a more discriminative topic model to be estimated, since with a very small λ, only the rarest words in a document will be taken as from the topic model. The influence of this prior is controlled by the confidence parameter μ. A larger μ would cause all the documents to have nearly identical λ's, whereas a smaller μ would allow us to discount different feedback documents more aggressively and differently. A smaller μ would also make the performance insensitive to λ_{prior}, since the prior would be weak. As will be discussed later, the experiment results do show that a smaller μ is indeed beneficial.

While this model appears to be similar to the model proposed in [8], there are two important differences:

1. In the old mixture model proposed in [8], we pool all the feedback documents together, and have a single mixture model for the "concatenated feedback document", whereas in our new model, each document has a *separate* mixture model, which allows us to model the different amount of relevant information in each document. As a result, we can discount documents with little relevant information in a principled way.
2. In the old mixture model, we do not consider the original query model when estimating the mixture model parameters, whereas in the new model, we use a Maximum A Posterior (MAP) estimator and use the original query model to define a conjugate prior for the topic language model θ_T. Although, in effect, this also results in a linear interpolation between the original query model and the feedback document model, the interpolation coefficient is dynamic and the query model can regulate the estimation of the feedback model, making the estimate more robust against shifting to a distracting topic in the feedback document set.

These two differences allow the new model to address the above-mentioned two problems with the old mixture model. Note that if we put an infinitely strong prior on all the λ_i's, and we do not use the original query model as a prior, we will recover the old model proposed in [8] as a special case.

4 Experiment Results

In this section, we present some experiment results with the proposed mixture model. Since the purpose of the mixture model is to estimate a potentially better query model θ_T by exploiting biased clustering of words in the feedback documents, we evaluate the effectiveness of our model and parameter estimation based on the retrieval performance from using the estimated query model θ_T.

4.1 Experiment Design

We use the Associated Press (AP) data available through TREC [7] as our document database, which has 164597 news articles. We use TREC topics $101 - 150$ as our experiment queries [7]. On average, each topic has about 100 relevant documents. We use the standard average precision measure to evaluate retrieval performance [1]; the average precision is a single number measure for a ranking result.

4.2 Experiment results

Influence of λ

The first research question we want to answer is whether the flexibility of allowing each document to have a different relevance parameter λ actually leads to a more accurate estimate of the query model. This question can be answered by varying the parameter μ while fixing all the other parameters.

As discussed in Section 3, the parameter μ is our confidence on the prior λ_{prior}. The larger μ is, the more we trust the prior. When μ goes to infinity, we essentially fix the λ of each document to a constant value λ_{prior}. On the other hand, when μ is set to zero, the model has maximum flexibility to allow each document to have a different λ. Therefore, by changing the value of λ, we can see how such flexibility affects the retrieval performance of the estimated query model. To exclude the influence of other parameters, we set $\sigma = 0$, which is equivalent to ignoring the query model prior entirely, thus the estimated query model is entirely based on the feedback documents.

The results are shown in Fig. 3, where the x-axis is different μ in log scale and the y-axis is the mean average precision over all the 50 topics. The two curves in the figure correspond to using 100 and 300 feedback documents, respectively.

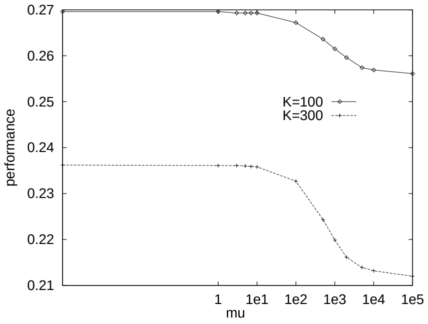

Fig. 3. Influence of μ .

From the results in Fig. 3, we see clearly that the performance drops as μ increases. Since a larger μ means less flexibility in estimating a different λ for each document, we can conclude from these results that allowing each document to have a potentially different λ – a feature of our new mixture model as compared with the old model proposed in [8] – indeed helps improve performance. Intuitively, this also makes sense, since with a small μ, the EM algorithm has more flexibility to estimate a potentially different λ for each document, which, in effect, achieves a weighting of each document when pooling the word counts to estimate the new query model. When μ is large, we do not have this flexibility, and *all* feedback documents are treated equally, which is not reasonable because not all the feedback documents are relevant. This is especially true when we use a large number of documents for feedback. These results are quite encouraging, as it suggests that we can simply set $\mu = 0$ and the model will be insensitive to the prior λ_{prior}. Compared with the old mixture model, this is a significant advantage, as in the old model, we must manually tune the parameter λ to optimize the retrieval performance [8].

Influence of query prior

The second question we want to answer is whether our treatment of the original query model as a prior has any advantages over the heuristic interpolation of the original query model with a feedback model as in [8]. This question can be answered by varying the parameter σ while fixing λ (to 0.9 and μ as infinity in our experiments).

Since σ reflects our confidence on the query model prior, it essentially controls how much weight we put on the original query model when mixing it with the new relevance information from the feedback document. Setting $\sigma = 0$ would ignore the original query model completely, while setting σ to a very large number would, in effect, turn off feedback and our estimated new query model would be precisely the original query model. One possible advantage of treating the original query model as a prior is that it allows a flexible interpolation in the sense that different queries may have a different interpolation coefficient, depending on how much relevance information exists in the feedback documents. This means that the optimal setting of σ can be expected to be more *stable* than that of the interpolation coefficient in the old mixture model.

Fig. 4 (left) shows the results of varying σ. We see that: (1) The optimal performance is usually achieved when σ is somewhere not too small and not too large, suggesting that a good balance between query prior and feedback relevant model is necessary to achieve optimal performance. (2) The optimal setting of σ appears to be insensitive to the number of documents used for feedback.

We also plotted how the performance is affected by the interpolation coefficient in the old mixture model in Fig. 4 (right). The two plots have similar patterns. Theoretically, when σ goes to 0, it is equivalent to that α goes to 1, when both methods perform worst. However, our method tends to be more flat/stable when σ increases than the old model when α goes to 0.

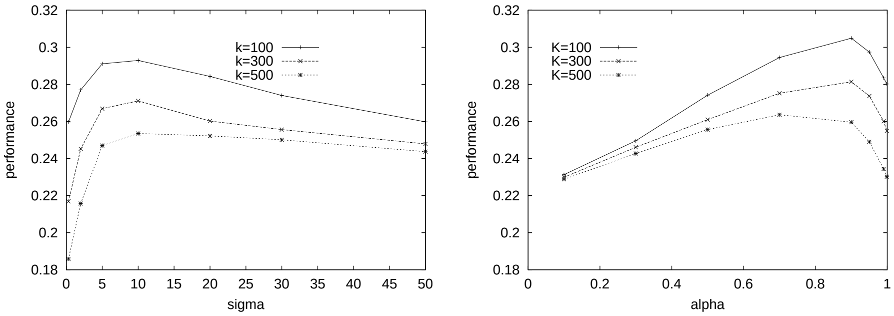

Fig. 4. Influence of σ for the new mixture model (left) and α for the old mixture model .

Sensitivity on the number of feedback documents

Finally, we examine the sensitivity of performance to the number of feedback document. For this purpose, we control σ and μ and vary the number of feedback documents. The results are plotted in Fig. 5 (left), where μ is set to infinity and λ is set to 0.9). For comparison,we also plot similar results from using the old mixture model in Fig. 5(right), where λ is set to 0.9. We see that in both figures, the performance is least sensitive to the number of documents when the original query model has the largest weight ($\sigma = 50$ for the new model and $\alpha = 0.1$ for the old model), but their absolute retrieval performance is not good, as we barely update the query model with the feedback documents. As we do more feedback, we see that the sensitivity pattern appears to be similar for both our new model and the old model. This suggests that while we allow each document to have a different λ, it does not penalize the low-ranked documents sufficiently to ensure the estimated model to be mainly based on the top ranked documents. This can also be seen from the fact that the performance tends to peak around using 50 documents for all parameter settings and for both models. That is, the number of documents to use is still the major parameter to set empirically.

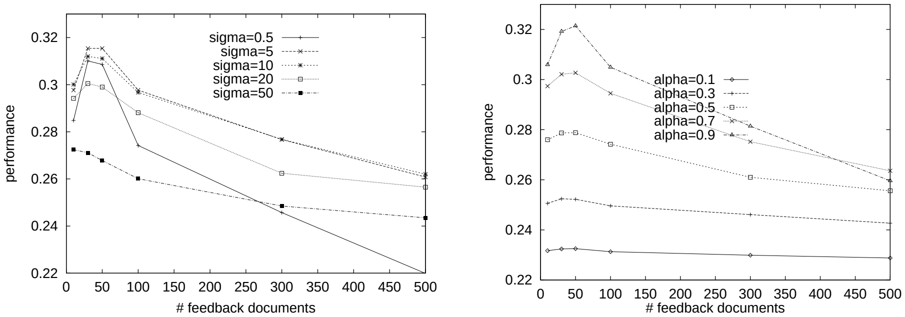

Fig. 5. Precision of using different number of feedback documents for new model (left) and old model(right)

5 Conclusions and further work

In this paper, we present a new mixture model for performing pseudo feedback for information retrieval. The basic idea is to treat the words in each feedback document as observations from a two-component multinomial mixture model, where one component is a topic model anchored to the original query model through a prior and the other is fixed to some background word distribution. We estimate the topic model based on the feedback documents and use it as a new query model for ranking documents. This new model extends and improves a similar existing mixture model in two ways: (1) It allows each feedback document to have a different mixing parameter, which is shown to improve retrieval performance in our experiments. (2) It incorporates the original query model into the mixture model as a prior on the topic model, which is a more principled way of updating the query model than the heuristic interpolation used in the old mixture model. Our model can be regarded as performing a "biased clustering" of words in the feedback documents.

There are several directions for further extending the work presented here. First, we need to test the model with more data sets to see if the patterns reported here are general patterns of the model. Second, the high sensitivity to the number of feedback documents remains an unsolved issue. It is very important to further improve the mixture model and the estimation methods to make it more robust against the change in the number of feedback documents. One possibility is to let the EM algorithm start with the original query model and *very conservatively* "grow" the topic model by incorporating the relevance information from the feedback documents. As long as we grow the topic model *"slowly"* and maintain a discrimination among documents based on relevance, the estimation can be expected to be more robust against the number of feedback documents. The growth of the topic model can be controlled by the two prior confidence parameters (for the topic model prior and the mixing weight prior). So one heuristic way for regulating the EM algorithm is to *dynamically* change these confidence parameters.

References

1. R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley, 1999.
2. W. B. Croft and J. Lafferty, editors. *Language Modeling and Information Retrieval.* Kluwer Academic Publishers, 2003.
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. J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR'01*, pages 111–119, Sept 2001.
5. J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR'98*, pages 275–281, 1998.
6. K. Sparck Jones and P. Willett, editors. *Readings in Information Retrieval.* Morgan Kaufmann Publishers, 1997.
7. E. Voorhees and D. Harman, editors. *Proceedings of Text REtrieval Conference (TREC1-9).* NIST Special Publications, 2001. http://trec.nist.gov/pubs.html.
8. C. Zhai and J. Lafferty. Model-based feedback in the KL-divergence retrieval model. In *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 403–410, 2001.
9. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*, pages 334–342, Sept 2001.