Improve Retrieval Accuracy for Difficult Queries using Negative Feedback

Xuanhui Wang
University of Illinois at Urbana-Champaign
Urbana, IL 61801
xwang20@cs.uiuc.edu

Hui Fang
The Ohio State University
Columbus, OH 43210
hfang@cse.ohio-state.edu

ChengXiang Zhai
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@cs.uiuc.edu

ABSTRACT

How to improve search accuracy for difficult topics is an under-addressed, yet important research question. In this paper, we consider a scenario when the search results are so poor that none of the top-ranked documents is relevant to a user's query, and propose to exploit negative feedback to improve retrieval accuracy for such difficult queries. Specifically, we propose to learn from a certain number of top-ranked non-relevant documents to rerank the rest unseen documents. We propose several approaches to penalizing the documents that are similar to the known non-relevant documents in the language modeling framework. To evaluate the proposed methods, we adapt standard TREC collections to construct a test collection containing only difficult queries. Experiment results show that the proposed approaches are effective for improving retrieval accuracy of difficult queries.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Retrieval models

General Terms: Algorithms

Keywords: negative feedback, language modeling, difficult queries

1. INTRODUCTION

Due to inherent limitations of current retrieval models, it is inevitable that some queries are difficult in the sense that the search results would be poor. Indeed, some queries may be so difficult that a user can not find any relevant document in a long list of top-ranked documents even if the user has reformulated the queries several times. Clearly, how to improve the search accuracy for such difficult queries is both practically important and theoretically interesting.

In this paper, we consider the scenario when the search results are so poor that none of the top-ranked documents is relevant to a user's query. In such a scenario, the feedback information that a user could provide, either implicitly or explicitly, is all negative. An interesting question is thus how to exploit only non-relevant information to improve search accuracy, which is referred to as *negative feedback*.

CIKM'07, November 6–8, 2007, Lisboa, Portugal.

Although several kinds of feedback, including relevance feedback, pseudo feedback, and implicit feedback, have been extensively studied in information retrieval, most existing work on feedback relies on positive information, i.e., exploiting relevant documents or documents that are assumed to be relevant. The basic idea is generally to extract useful terms from positive documents and use them to expand the original query or update the query model. When positive documents are available, they are generally more useful than negative documents [1]. As a result, how to exploit negative documents for feedback has been largely under-addressed. The only work that we are aware of is query zone [6]. But this work is in the context of document routing where many relevant documents are available. In contrast, we focus on feedback based solely on negative documents in the context of ad hoc search.

Indeed, when a query is difficult, it is often impossible to obtain positive documents for feedback. Thus the best we could do is to exploit the negative documents to perform negative feedback. In this paper, we study negative feedback in the language modeling retrieval framework. Our basic idea is to identify the distracting non-relevant information from the known negative example documents, and penalize unseen documents containing such information. While this idea is naturally implemented in a traditional feedback method such as Rocchio [5], we show that it cannot be naturally implemented in the language modeling approach. We thus propose a heuristic implementation of this idea in the language modeling approach in a similar way to how it is implemented in Rocchio. Specifically, we would first estimate a negative topic model based on the negative example documents, and then combine this negative model with the original query model to penalize documents whose language models are similar to the negative topic model. We further propose two strategies to improve this basic negative feedback method: First, we propose to down-weight or eliminate query terms in the learned negative model. The idea is so that the learned negative model would be focused on the truly distracting aspects rather than the query related aspects in a non-relevant document. Second, we propose to model multiple possible distracting negative subtopics in the negative examples documents, so that we can penalize a document as long as it is similar to one non-relevant document or one non-relevant aspect.

To evaluate the effectiveness of the proposed methods, we construct a test collection containing only difficult queries from TREC collections. Experiment results show that the proposed basic negative feedback method is effective for improving ranking accuracy of difficult queries, and query term elimination can help further improve ranking effectiveness. However, modeling multiple negative models does not show a clear gain and further investigation is needed.

2. PROBLEM FORMULATION

Given a query Q and a document collection $\mathcal{C}$, a retrieval system returns a ranked list of documents $\mathcal{L}$. We use L_i to represent the document at i-th position in the ranked list.

We focus on the scenario when a query is very difficult such that a user cannot find any relevant document in the top f ($f = 10$ in this study) ranked documents. In this case, the user can provide the retrieval system with negative feedback information either explicitly or implicitly (e.g., skipping these f documents or clicking on the "Next" button). Our goal is to study how to use these negative example documents, $N = \{L_1, ..., L_f\}$, to effectively rerank the next r ($r = 1000$ in this study) *unseen* documents in the original ranked list: $U = \{L_{f+1}, ..., L_{f+r}\}$.

Our problem setup can be regarded as a special case of relevance feedback, where all the feedback information is negative. However, since no positive example is assumed to be available, our problem is much more challenging than regular relevance feedback.

3. LANGUAGE MODELING APPROACHES TO NEGATIVE FEEDBACK

In this section, we study the problem of negative feedback in the language modeling framework. We first review the language modeling approach and discuss the difficulty of incorporating negative feedback information in any truncated query model. To overcome this difficulty, we then propose to use a heuristic method to incorporate negative feedback through language modeling.

3.1 Language Models to Information Retrieval

In the basic language modeling approach [4], we score a document D by the likelihood of generating query $Q = (q_1, ..., q_m)$ from a document language model θ_D. That is, we first estimate a multinomial distribution θ_D for D and then compute

$$p(Q|D) = \Pi_{i=1}^{m} p(q_i|\theta_D).$$

The document model θ_D needs to be smoothed and an effective method is Dirichlet smoothing [9]:

$$p(w|\theta_D) = \frac{c(w, D) + \mu p(w|\mathcal{C})}{|D| + \mu}$$

where $c(w, D)$ is the count of word w in D, $|D|$ is the length of D, $p(w|\mathcal{C})$ is the collection language model, and μ is a Dirichlet smoothing parameter. This smoothing method is what we will use in our experiments.

The above query likelihood method is quite related to the KL-divergence retrieval model in [3]. In KL-divergence model, a query model θ_Q is also estimated for a query Q. Then a document D is ranked based on the KL-divergence between θ_Q and θ_D

$$-D(\theta_Q||\theta_D) = -\sum_{w \in V} p(w|\theta_Q) \log \frac{p(w|\theta_Q)}{p(w|\theta_D)}$$

where V is the set of words in our vocabulary.

Using the maximum likelihood estimation of θ_Q, it can be shown that ranking based on the KL-divergence is equivalent to ranking based on the query likelihood [3]. Therefore, query likelihood can be regarded as a special case of the KL-divergence method.

Indeed, one major motivation for the KL-divergence retrieval model is to address the difficulty in incorporating feedback into the query likelihood method [8]. In the KL-divergence model, one can cast feedback as updating the query language model. Unfortunately, while it is easy to incorporate feedback with positive documents, it cannot naturally accommodate negative feedback. The reason is because with a query model, in which every term has a *non-negative* probability, it is difficult to penalize a term without including all other terms. Specifically, if we are to penalize a term, the best we could do is to assign a very small or zero, but non-negative probability to it. Unfortunately, in a truncated query model (i.e., only keeping the most significant terms and assuming all others to be zero), a small non-zero probability for a distracting term (i.e., a term to be penalized) actually means that the term would be favored more than many non-distracting terms that aren't in the truncated query model as the latter would all have zero probability. In order to penalize such distracting terms, we would have to include *all* other terms with higher probabilities than these distracting terms. Thus it is very difficult to incorporate negative information by updating query model and we need some new negative feedback methods in the language modeling approaches.

3.2 KL-divergence for Negative Feedback

Since neither the query likelihood method nor the KL-divergence method can naturally support negative feedback, we propose a heuristic modification of the KL-divergence retrieval method to perform negative feedback. Intuitively, in negative feedback, we would like to push down the documents that are similar to the known negative example documents. Following the spirit of language models, one way to do this would be to estimate a *negative topic language model* θ_N. We could then use θ_N to retrieve documents that are potentially distracting and compute a "distraction score" for each document. The distraction score of a document can then be combined with the original KL-divergence score of the document in such a way that we would penalize a document that has a high distraction score.

Specifically, let θ_Q be the estimated query model for query Q and θ_D be the estimated document model for document D. Let θ_N be a negative topic model estimated based on the negative feedback documents $N = \{L_1, ..., L_f\}$. For negative feedback, we would score D with respect to Q as follows:

$$Score(Q, D) = -D(\theta_Q||\theta_D) + \beta \cdot D(\theta_N||\theta_D) \qquad (1)$$

where β is a parameter that controls the influence of negative feedback. When $\beta = 0$, we do not perform negative feedback, and the ranking would be the same as the original ranking according to θ_Q. We call this method the basic negative feedback model (BasicNFB) to distinguish it from some other extensions that we will propose later.

After some simple algebra transformation and ignoring those constants that do not affect ranking of documents, it is easy to show that Equation (1) can be rewritten as:

$$\begin{aligned} Score(Q, D) &= -D(\theta_Q||\theta_D) + \beta \cdot D(\theta_N||\theta_D) \\ &\overset{\text{rank}}{=} \sum_{w \in V} [p(w|\theta_Q) - \beta \cdot p(w|\theta_N)] \log p(w|\theta_D) \end{aligned}$$

In this new form of the BasicNFB scoring formula, we see that each term has a weight of $[p(w|\theta_Q) - \beta \cdot p(w|\theta_N)] \log p(w|\theta_D)$, which penalizes a term that has high probability in the negative topic model θ_N. Thus the BasicNFB model is essentially very similar to Rocchio in the vector space model [5] and can in some sense be regarded as the language modeling version of Rocchio. Clearly, our main question now is how to estimate θ_N, which will be discussed below.

3.3 Estimation of Negative Topic Model

Given a set of non-relevant documents $N = \{L_1, ..., L_f\}$, we would like to learn a negative topic language model θ_N from this set of documents. This is very similar to the case when we need to

perform positive feedback with positive example documents. Thus we can use the same mixture model as used in [8] for pseudo feedback to estimate θ_N.

Specifically, we assume that all the non-relevant documents are generated from a mixture of a unigram language model θ_N (to generate non-relevant information) and a background language model (to generate common words). As usual in language modeling, we use the collection language model $p(w|\mathcal{C}) = \frac{c(w,\mathcal{C})}{\sum_w c(w,\mathcal{C})}$ as the background model. Then the log-likelihood of the sample N is

$$L(N|\theta_N) = \sum_{d \in N} \sum_{w \in d} c(w,d) \log[(1-\lambda)p(w|\theta_N) + \lambda p(w|\mathcal{C})]$$

where λ is a mixture parameter which controls the weight of the background model. Given a fixed λ, a standard EM algorithm can then be used to estimate parameters $p_\lambda(w|\theta_N)$:

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

$$p_\lambda^{(n+1)}(w|\theta_N) = \frac{\sum_{d \in N} c(w,d)t^{(n)}(w)}{\sum_w \sum_{d \in N} c(w,d)t^{(n)}(w)}.$$

The result of the EM algorithm would give a discriminative negative model θ_N which eliminates background noise.

3.4 Query Term Elimination

The negative model learned above is based on the top documents returned to a query. This means all these documents may have high occurrences of query terms. As a result, the query terms would likely have high probabilities in the negative feedback model. This could make the negative feedback model biased and thus ineffective to identify those truely irrelevant documents. To address this problem, we propose to eliminate the query terms from the negative model by setting their probabilities to zero and name this technique as "query term elimination."

3.5 Multiple Negative Models

While positive example documents are generally coherent, negative feedback examples may be quite diverse as different negative documents may distract in completely different ways. Thus a single negative topic model may not be optimal. In this section, we propose to estimate multiple negative models and use them to perform negative feedback.

A principled way of achieving multiple negative models is to learn subtopics from the negative documents. We use the modified probabilistic latent semantic analysis (PLSA) model [2] proposed in [10] to estimate k topics from N, each corresponding to a unigram langauge model $\{\theta_i : 1 \leq i \leq k\}$.

To compute a distraction score of a document D with multiple negative topics,we compute the KL-divergence of θ_D and each of the negative models θ_i, and choose the minimum divergence (i.e., highest similarity) as the distraction score of the document. This distraction score is then combined with the original KL-divergence as in BasicNFB. That is,

$$Score(Q,D) = -D(\theta_Q||\theta_D) + \beta \min\{D(\theta_i||\theta_D) : 1 \leq i \leq k\}.$$

Note that a special case of this method is to have each document as a cluster and let each document define its own negative model.

Directly implementing the multiple negative models is quite expensive since we have to find a minimum negative model for every document. An efficient way is to do it reversely. That is, for each negative model θ_i, we use the KL divergence model to rank all the documents in the collection. We then select the top n documents for each negative model and form a pool set D_P. For each document $d \in D_P$, we compute $Score(Q,d)$ using the formula above. For any other document $d' \notin D_P$, we simply score d' as $-D(\theta_Q||\theta_{d'}) + \beta \cdot c$, where $c = \max_{d \in D_P} \min_{1 \leq i \leq k} D(\theta_i||\theta_d)$. Note that the query term elimination technique can also be used with multiple negative models.

4. EXPERIMENTS

We set up our experiments to simulate the following real-world scenario of difficult queries. After submitting a query to a search engine, a user would go through the top 10 ranked documents, but find that none of them is relevant, so the user would click on the "Next" button to view the second result page. Our goal is to improve the ranking accuracy of unseen results after the user clicks on the "Next" button. Specifically, given a baseline retrieval method, we identify the top 10 ranked documents and treat them as already seen by the user. We then exclude them and study different feedback techniques to re-rank the remaining 1,000 unseen documents.

We use the TREC 2004 ROBUST track document collection, which contains approximately 528,000 documents [7]. Since negative feedback is meant to help difficult topics, in our evaluation, we use only the relatively difficult topics from the 249 queries used in the ROBUST track of TREC 2004 [7]. Specifically, we choose the topics with low precision at 10 documents (P@10) according to a baseline method (the KL-divergence model with Dirichlet smoothing [9]). Based on such a criterion, we created two query sets:

Hard1: 51 queries, for which, the baseline system returned at most 1 relevant document in top 10 and at most 3 relevant documents in top 20 documents. To make the topic fit our evaluation setup, we remove all relevant documents in the top 10 results as if they were not existing in our collection.

Hard2: 26 queries, for which, the baseline system failed to return any relevant document in the top 10 ranked documents.

We use two sets of queries because they complement each other in the sense that Hard2 better reflects the real scenario while Hard1 has more queries to experiment with. For both query sets, we preprocess documents and queries with stemming, but without removing any stopword.

We use two sets of performance measures: (1) MAP and gMAP, which serve as good measures of the overall ranking accuracy. (2) MRR, and P@10, which reflect the utility for users who only read the very top ranked documents. Due to the space limit, we only show the optimal results for different methods after tuning the parameters (we ended up setting $\beta = 0.5$ and $\lambda = 0.8$).

4.1 Overall Performance Comparison

We compare the optimal performance of our proposed methods with the baseline and traditional model-based feedback method [8]. We use the following notations: **BL** is KL-divergence with Dirichlet smoothing [9]. **PFB** is the model-based feedback method [8]. **BasicNFB** is the proposed basic negative feedback model. **QTE** represents query term elimination technique. **MNFB-single** represents the proposed multiple negative models using single documents as non-relevant aspects. **MNFB-cluster** represents the proposed multiple negative models using clustering method and we set the number of clusters to 3.

The results are shown in Table 1. We can see that the observations on these two query sets are mostly consistent. It is clear that traditional expansion-based method (PFB) can not improve the retrieval performance for these difficult queries; instead, they hurt the

Table 1: Overall Performance Comparison

Methods	**Hard1** (51 queries)				**Hard2** (26 queries)			
	MAP	gMAP	MRR	P@10	MAP	gMAP	MRR	P@10
BL	0.0405	0.0188	0.221	0.0686	0.0294	0.0138	0.148	0.076
PFB	0.0377	0.0186	0.161	0.0765	0.0275	0.0120	0.147	0.048
BasicNFB+QTE	**0.0470**	**0.0199**	**0.252**	**0.08**	0.0329	0.0147	0.211	**0.088**
MNFB-single+QTE	0.0462	0.0192	0.232	0.0745	0.0352	**0.0148**	0.183	0.076
MNFB-cluster+QTE	0.0457	0.0191	0.243	0.0765	**0.0364**	0.0144	**0.213**	0.076

Table 2: Effectiveness of Query Term Elimination

Methods	**Hard1** (51 queries)				**Hard2** (26 queries)			
	MAP	gMAP	MRR	P@10	MAP	gMAP	MRR	P@10
BasicNFB	0.0468	0.0195	0.234	0.0784	0.0353	0.0144	0.204	0.076
BasicNFB+QTE	0.0470	0.0199	0.252	0.08	0.0329	0.0147	0.211	0.088
MNFB-single	0.0454	0.0193	0.236	0.0745	0.0354	0.0148	0.189	0.076
MNFB-single+QTE	0.0462	0.0192	0.232	0.0745	0.0352	0.0148	0.183	0.076
MNFB-cluster	0.0453	0.0193	0.259	0.0745	0.0354	0.0145	0.199	0.076
MNFB-cluster+QTE	0.0457	0.0191	0.243	0.0765	0.0364	0.0144	0.213	0.076

performance substantially. This should not be surprising since PFB blindly takes non-relevant information as relevant to update query models. On the contrary, our proposed methods are more effective for negative feedback and outperform both the baseline and traditional pseudo-feedback method. This confirms our hypothesis that negative feedback does help improve the accuracy. Compared with BasicNFB, multiple negative models (MNFB-single and MNFB-cluster) are more effective on the Hard2 query set but less effective on Hard1 query set. Overall, multiple negative models do not show a clear gain over the basic negative model. One possible reason is that only 10 documents can not give a reliable cluster structures due to the local maximum problem of the clustering method.

4.2 Effectiveness of Query Term Elimination

We examine the effectiveness of the proposed query term elimination in Table 2. For BasicNFB, it is clear that query term elimination helps improve the performance on both query sets for all measures except for MAP on Hard2, in which case, the MAP value of BasicNFB+QTE is lower mainly because BasicNFB+QTE did poorly on one single topic (topic 320), causing the arithmetic mean to be lower. However, for MNFB-single and MNFB-cluster, query term elimination appears to be ineffective. One possible reason is because the basic negative model is learned from *all* the top results of a query, which presumably have a high concentration of query terms, thus the query terms would be more salient in the learned negative model than in the case of multiple negative model methods, where the negative model is learned only from one single document or subtopic/cluster.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we show that when a query is difficulty and the search results are so poor that none of the top-ranked documents is relevant to a user's query, we may exploit negative feedback to improve retrieval accuracy. Performing negative feedback with the language modeling approach is non-trivial. We propose a KL-divergence approach to negative feedback which is in spirit similar to Rocchio for the vector space model [5] with the main idea being to penalize those documents that are similar to the known non-relevant documents. To evaluate the proposed methods, we adapt standard TREC collections to construct a test collection containing only difficult queries. Experiment results show that the proposed approaches are effective for improving the retrieval accuracy of difficult queries.

To the best of our knowledge, our work is the first to study feedback with only non-relevant documents. There are some natural future research directions based on this work, including investigating how to automatically set the parameters and developing a more principled way to do negative feedback in the language modeling framework.

6. ACKNOWLEDGMENTS

This work is in part supported by the National Science Foundation under award numbers IIS-0347933 and IIS-0713581.

7. REFERENCES

[1] M. Dunlop. The effect of accessing non-matching documents on relevance feedback. *ACM TOIS*, 15(2), 1997.

[2] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[3] J. D. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR*, pages 111–119, 2001.

[4] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[5] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval SystemąłExperiments in Automatic Document Processing*, pages 313–323, 1971.

[6] A. Singhal, M. Mitra, and C. Buckley. Learning rounting queries in a query zone. In *SIGIR*, pages 25–32, 1997.

[7] E. M. Voorhees. Draft: Overview of the trec 2005 robust retrieval track. In *Notebook of the Thirteenth Text REtrieval Conference (TREC2005)*, 2005.

[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, 2001.

[10] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text minning. In *Proceeding of the 10th ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 743–748, 2004.