Interactive Construction of Query Language Models – UIUC TREC 2005 HARD Track Experiments

Bin Tan[1], Atulya Velivelli[2], Hui Fang[1], ChengXiang Zhai[1]

[1]Department of Computer Science
University of Illinois at Urbana-Champaign

[2]Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

1 Introduction

In the language modeling approach, feedback is often modeled as estimating an improved query model or relevance model based on a set of feedback documents [3, 1]. This is in line with the traditional way of doing relevance feedback – presenting the user with documents or passages for relevance judgment and then extracting terms from the judged documents or passages to improve a query model. Such an indirect way of obtaining help from a user to construct a query model has the drawback that irrelevant terms that occur with relevant ones in the judged content may be erroneously used for query model modification.

A more direct way to involve a user in improving the query model is to present some candidate terms to a user and directly ask the user to judge the relevance of each term or specify the probability of each term. This strategy has been discussed in [2], but has not been seriously studied in any existing work. In participating in the TREC 2005 HARD Track task, we explored how to exploit term-based feedback to better involve a user in constructing an improved query model for information retrieval with language models.

There are several research problems related to the "term feedback" approach in general. First, since the user is only likely to judge the relevance of a modest number of terms, it is important to study which terms to present to the user for judgment. If the query is ambiguous, i.e., there are multiple clusters of the retrieved documents and the user is only interested in one of them, it is necessary to diversify the presented terms so that each cluster gets adequate representation. It would be unwise to devote the presentation completely to terms from a dominant cluster, because if the user happens to be searching for documents in a small cluster, then he/she gets no chance to indicate his/her interests as the small cluster is under-represented.

Second, we must design a weighting scheme to designate how good a term is for relevance feedback. It is usually not sufficient to only assign weights to terms presented for judgment, since the number of these terms are not many. If an unpresented term is similar to a presented term, then there is good reason to propagate weight of the presented term to the unpresented one. Also, it is tricky how to treat negative term feedback. A presented term may be left unselected because it is irrelevant, or because the user does not have enough knowledge to determine its relevance.

Third, a serious drawback of term feedback is that, unlike document or passage feedback, the presented terms have almost no context, which makes it hard for a user to judge their relevance. If a term is only meaningful when combined with other terms or if the term is part of a name or terminology which the user has not heard of, then it is very likely to be missed by the user.

In our exploration, we address the first two problems of term relevance feedback. Specifically, we use clustering to nominate representative terms and construct the query language model based on both judged terms and their attached clusters. Experiment results show that our approach is effective for involving a user to interactively construct an accurate query language model based on term judgments.

2 Retrieval Method

We use the KL-divergence retrieve method, in which the relevance score of a document with respect to a query is given by the KL-divergence value between the document language model θ_D and the query language model θ_Q. If a feedback language model θ_F can be estimated from either pseudo-feedback or relevance feedback, a modified query model $\theta_{Q'} = \lambda\theta_Q + (1-\lambda)\theta_F$ can be used in place of θ_Q [3]. We use Lemur's implementation of the KL-divergence retrieval method with Dirichlet prior smoothing (parameter set to 2000).

3 Methods for Query Model Construction

For each query topic, the collection of 50 top-ranked result documents returned by the KL-divergence retrieval method is taken as the topic context from which to choose feedback terms. We use the mixture multinomial distribution model method proposed in [4] to discover K theme clusters within this small query-specific collection. Each cluster represents a possible latent theme in the collection and is simply a multinomial word distribution (also called unigram language model).

We select L distinct terms with highest probabilities from each cluster to form a pool of $K \times L$ terms. If a term is suggested by different clusters, then it is assigned to the one in which it has highest probability. Terms that occur in the title of the topic are filtered. We then present these terms in a clarification form to the user for evaluation of their relevance to the topic.

Before discussing the details our methods, we introduce some notations:

- θ_Q: The query language model as mentioned in the previous section.
- θ_F: The feedback language model as mentioned in the previous section.
- $\theta_i\,(i = 1 \ldots K)$: The unigram language model of cluster i.
- $p(w|\theta_i)$: Probability of term w in θ_i.
- $p(\theta_i)$: Prior probability that a document is sampled from θ_i.
- $T = \{t_{i,j}\}\,(i = 1 \ldots K, j = 1 \ldots L)$ The set of terms presented to the user in the clarification form. $t_{i,j}$ is the j-th term chosen from cluster i.
- δ_w: Indicator variable which is 1 if term w is in T and judged relevant by the user and 0 otherwise.

Given the evaluation results $\delta_{i,j}$, the task is to compute a feedback language model θ_F which reflects the user's judgment that all terms for which $\delta_{i,j} = 1$ are relevant to the topic. Hopefully, the feedback model θ_F (possibly interpolated with the query model θ_Q) will improve the retrieval accuracy.

We now describe several different methods for computing θ_F.

1. BLPFB (Baseline Pseudo-Feedback): This method does not use relevance judgments.

$$p(w|\theta_F) = \sum_{i=1\ldots K} p(w|\theta_i)p(\theta_i)$$

2. TFB (Term Feedback): We give non-zero weights to those terms that are judged relevant by the user. We interpolate it with the original query model and set Dirichlet prior to be proportional to query length $|Q|$ so that longer queries receive more weight. Cluster information (i.e., which cluster a judged term comes from) is not used.

$$p(w|\theta_F) = \frac{\delta_w + \mu p(w|\theta_Q)}{\sum_{w' \in T} \delta_{w'} + \mu},\ \mu = k|Q|$$

   For this method we do not need to interpolate θ_F with θ_Q, as θ_Q is already used in the computation of θ_F.

3. CFB (Cluster Feedback): Cluster language models are interpolated with weights determined by the number of presented terms that are judged relevant in each cluster. We do not distinguish which terms are judged relevant in a cluster; only the count matters.

$$p(w|\theta_F) = \sum_{i=1\ldots K} p(w|\theta_i) \frac{\sum_{j=1\ldots L} \delta_{t_{i,j}} + \nu p(\theta_i)}{\sum_{k=1\ldots K, j=1\ldots L} \delta_{t_{k,j}} + \nu}$$

4. TCFB (Term-Cluster Feedback): This is simply an interpolation of TFB and CFB:

$$p(w|\theta_F) = \lambda p(w|\theta_{F_{TFB}}) + (1-\lambda) p(w|\theta_{F_{CFB}})$$

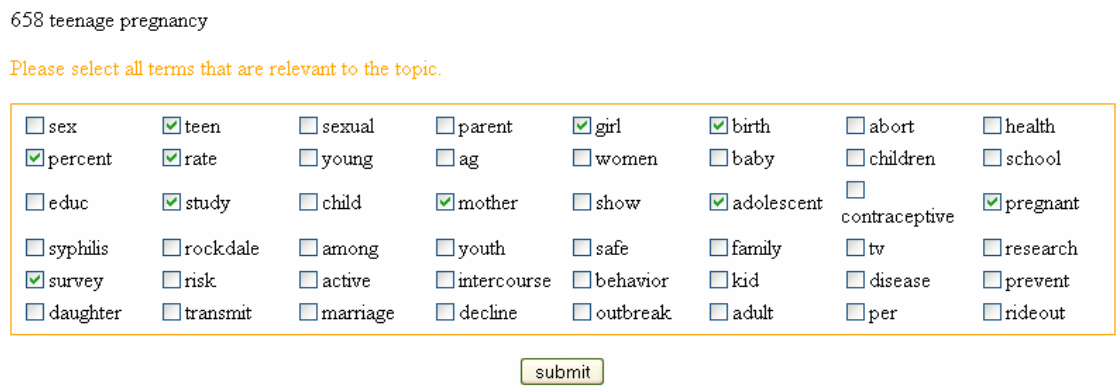

Figure 1: Filled clarification form for topic 658

As was in the case of TFB, there is no need to interpolate θ_F with θ_Q.

This method is proposed as we believe TFB and CFB have strengths and drawbacks in different aspects. TFB assigns weights to the presented terms but completely ignores unpresented terms. CFB remedies this by treating terms in a cluster equally, such that unpresented terms receive weights when presented terms in that cluster are selected, but it does not differentiate which terms in the cluster are selected. By combining the two methods we believe we can get the greatest benefits.

4 Clarification Forms

As described in the previous section, each clarification form (CF) contains terms from different clusters for relevance feedback. The evaluators were asked to select all terms that she/he deemed relevant to the topic but did not know that the terms form clusters. We generated three sets of clarification forms: 1×48, 3×16 and 6×8. The 1×48 CFs contain 48 terms from a single large cluster. The 3×16 CFs have 3 clusters and 16 terms from each. The 6×8 CFs have 6 clusters and 8 terms from each. Figure 1 shows an example of filled 3×16 clarification form. Table 1 shows some statistics about the number of selected terms in each set of clarification forms.

The trend that the average number of selected terms increases with the number of clusters seems to support our hypothesis that a larger number of theme clusters tend to diversify the presented terms so that the user gets a better chance to find terms that match the topic. However, when we evaluated the CFs generated for the HARD 2004 topics, the average numbers of selected terms were 18.4 for the 1×48 CF and 16.8 for the 6×8 CF, which seemed to contradict the hypothesis.

Table 1: Number of feedback terms in clarification forms

CF	Average	Std. Dev.
1×48	11.8	7.40
3×16	14.0	9.68
6×8	15.6	9.57

5 Experiment Results

Our baseline run (BLNFB) uses the KL-divergence language model based retrieval method with pseudo feedback (feedback documents = 5). For other runs we use the parameters that worked best when we evaluated the HARD 2004 topics. More specifically, we set μ to $4|Q|$ in TFB, ν to 0.000001 in CFB and λ to 0.8 in TCFB. The statistics of our runs are displayed in Table 2. The runs are tagged with the method and the number of clusters used. For example, TCFB6C denotes the run with

the TCFB method and 6×8 CFs.

Table 2: Mean average precision (MAP) and precision at 10 (Prec@10) for different runs. The ones marked by * are submitted official runs.

Run	MAP	Prec@10
BLNFB*	0.2244	0.460
TFB1C*	0.2890	0.546
TFB3C*	0.2929	0.556
TFB6C*	0.2816	0.542
CFB1C	0.2391	0.446
CFB3C	0.2966	0.548
CFB6C	0.2859	0.514
TCFB1C*	0.2898	0.540
TCFB3C*	0.3018	0.568
TCFB6C*	0.2917	0.530

Table 3: Performance of TFB3C under different settings of $\mu = k|Q|$.

k	MAP
0	0.2411
1	0.2687
2	0.2823
3	0.2890
4	0.2929
5	**0.2954**
6	0.2951
7	0.2944
8	0.2923
9	0.2914
10	0.2894

We find that the 3×16 runs yield the best performance, although the difference between the runs using different CFs is small.[1] This is again contrary to our hope that a larger cluster size (6×8) would facilitate better term feedback by suggesting more diversified terms. A close look of the results, however, reveals that the 3×16 and 6×8 runs excel at different topics, which means neither of them performs better than the other consistently. For example, the average precision of TCFB3C is higher than TCFB6C by more than 0.2 for as many as 13 topics, while the average precision of TCFB6C is higher by more than 0.2 for an equal number of other topics. We speculate that this may have something to do with how the true number of clusters matches the predefined number of clusters (either 3 or 6).

[1] CFB1C is an exception. Because it has only one cluster, it cannot incorporate term relevance feedback information by adjusting cluster weights.

Table 4: Performance of TCFB3C with $\mu = 5|Q|$ and different settings of λ. When $\lambda = 0$, it equals TFB3C. When $\lambda = 1$, it equals CFB3C.

λ	MAP
0	0.2954
0.1	0.3036
0.2	0.3081
0.3	0.3108
0.4	**0.3116**
0.5	0.3108
0.6	0.3093
0.7	0.3065
0.8	0.3030
0.9	0.2985
1	0.2966

For TFB3C, we vary the Dirichlet prior $\mu = k|Q|$ with k from 0 to 10. The results are shown in Table 3. The best retrieval performance is achieved when $k = 5$.

We also vary the interpolation factor λ in the TCFB3C calculation, the result of which is given in Table 4. The best value for λ is 0.4, which differs from the one we got on the HARD 2004 data.

The performance of TFB and CFB is comparable, and combining them with TCFB gives best performance for all three CFs. We look at how their performance differs across topics. Table 5 sums up the number of topics on which one methods is significantly better and worse than another method:

We find that TFB is more stable than CFB when compared to baseline, as it performs worse than the baseline for only 2 topics rather than 8 in the case of CFB. TFB and CFB perform better than each other in 11 topics respectively, which confirms our hypothesis that they have relative strengths. When combining the two methods with TCFB, we are able to get the best results by reducing the number of topics in which TCFB performs worse to any of them to 4.

Table 5: Number of topics on which one method is better and worse than another method by more than 0.05 in average precision. AP_1 is the average precision of Method 1. AP_2 is the average precision of Method 2.

Method 1	Method 2	$AP_1 - AP_2 > 0.05$	$AP_2 - AP_1 > 0.05$
TFB3C	BLNFB	22	2
CFB3C	BLNFB	24	8
TCFB3C	BLNFB	26	6
TFB3C	CFB3C	11	11
TFB3C	TCFB3C	4	12
CFB3C	TCFB3C	4	8

6 Conclusions

In this study we proposed several methods for utilizing term relevance feedback information to construct an improved query language model. We compare these methods and show how they can be combined to achieve best retrieval performance. There is still much to be explored in this direction of research. We should study better ways to incorporate both term and cluster feedback information. We should make best use of both positive and negative term feedback. It is also important to study how to present the terms in context to facilitate the user's relevance judgment.

References

[1] V. Lavrenko and B. Croft. Relevance-based language models. In *Proceedings of SIGIR'01*, pages 120–127, Sept 2001.

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

[3] 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.

[4] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of KDD*, 2004.