Extending Relevance Model for Relevance Feedback

Le Zhao, Chenmin Liang and Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
{lezhao, chenminl, callan}@cs.cmu.edu

Abstract

Relevance feedback is the retrieval task where the system is given not only an information need, but also some relevance judgement information, usually from users' feedback for an initial result list by the system. With different amount of feedback information available, the optimal feedback strategy might be very different. In TREC Relevance Feedback task, the system is given different sets of feedback information from 1 relevant document to over 40 judgements with at least 3 relevant. Thus, in this work, we try to develop a feedback algorithm that works well on all levels of feedback by extending the relevance model for pseudo relevance feedback to include judged relevant documents when scoring feedback terms. Within these different levels of feedback, it is more difficult for the feedback algorithm to perform well when given minimal amount of feedback. Experiments show that our algorithm performs robustly in those difficult cases.

1 Introduction

Relevance feedback is the retrieval task where the system is given not only a user query, but also user feedback on some of the top ranked results. Feedback gives the retrieval system a chance to improve its results by exploiting the extra information through more elaborate techniques. This can be helpful in cases where the users want as many relevant results as possible. Experimenting with and failure analyses on feedback algorithms also provides us a chance to understand what are missing in users' keyword queries and how to improve these queries.

In the following, we first introduce the relevance feedback task defined by this year's track, the basis ad hoc retrieval algorithm. In Section 2, we layout the basic model for combining pseudo relevance feedback and true relevance feedback in query expansion. One difficulty in doing that is properly and effectively balancing term weights from the two sources. We give our solution. We show experiments in Section 3 and conclude.

1.1 Task definition

With this year's relevance feedback experiments, the retrieval system will be evaluated on different number of feedback documents. Specifically, for each query, set A is the baseline without any feedback information. Set B allows the system to use 1 relevant document as positive feedback; set C allows 3 relevant and 3 non-relevant; set D uses 10 judged documents (a superset of set C), and set E provides all judged documents to the retrieval algorithm, varying from 40 to 100s of judged documents.

1.2 Pseudo relevance feedback as basis algorithm

Since different numbers of feedback documents are possible, especially, when only given a small number of judged documents, any relevance feedback algorithm that only look at the few judged documents can have difficulty providing reliable expansion queries. This is because fewer feedback documents means larger variations of term appearances and expansion weights. The approach we propose here, is to leverage the small amount of feedback information with some pseudo relevance feedback, so that the larger number of feedback documents provide lower variance estimates of the feedback terms, stabilizing the performance of the feedback algorithm.

The motivation for using the particular pseudo relevance feedback method proposed in (Lavrenko and Croft 2001) is that the best runs on the two previous TREC 2006 and 2007 Terabyte tracks, (Metzler et al 2005) and (Li and Yan 2006), both used this particular pseudo relevance feedback method. Since the relevance feedback task uses the same collection and similar set of queries, we set the baseline retrieval method to be the same as in the two previous top runs – a dependency model to achieve higher top precision and a pseudo relevance feedback step following.

Because of its relevance to this work, we show the pseudo relevance feedback model implemented in the Indri search engine:

$$P(r \mid I) = \frac{\sum_{D \in \text{topN}} P(r \mid D) P(I \mid D) P(D)}{P(I)} \qquad (1)$$

Relevance_model_query:

#weight(w_1 r_1 w_2 r_2 .. w_n r_n), where, $w_i = P(r_i \mid I)$.

Final_query:

#weight((1 - w) Relevance_model_query
w Original_query)

Here, r is a concept term, which could refer to any term or phrase (we restrict ourselves to only keywords), D refers to a feedback document, and I means the information need which is expressed by the query. $P(r \mid I)$ is a multinomial distribution over all concepts, of how ideally the relevant class should look like.

$P(I \mid D)$ is approximated with the probability for a document to generate the query, which is just the language modeling version of the relevance score. As $P(I)$ is constant w.r.t. r, $P(r \mid I)$ is easily calculated if we assume $P(D)$ is uniform. Top weighted terms are selected for query expansion and weighted exactly as their weights, so that the expansion query ranks documents according to their KL-divergence to the relevance model $P(r \mid I)$. The final query is just interpolating the expansion query with the original query. The original queries can be the keyword query titles provided by TREC or the dependency model phrase queries which we used as basis algorithm.

Now we introduce our relevance feedback algorithm.

2 Relevance feedback algorithm

The design of a feedback algorithm would be easy if the system is given complete judgments for top N results, just use the judged documents as training samples, use simple term features and train a binary classifier for relevant or not, see for example (Zhu, Zhao and Callan 2007). This will be especially effective if N is large. However, in many cases the system does not have such a luxury. In the current track, 3 out of the 4 feedback sets have N as low as 1, 6 or 10.

Thus, in this section we propose a unified pseudo-relevance and true relevance feedback method based on the relevance model formalism (Lavrenko and Croft 2001).

From equation (1) using Bayes rule, we get the original form of equation (1),

$$P(r \mid I) = \sum_{D \in \text{feedback}} P(r \mid D) P(D \mid I) \qquad (2)$$

For the feedback documents that have relevance judgments, i.e. $D \in D_T$, we trust the judgments, thus we set $P(D \mid I) = \text{Rel}(D, I) \big/ \sum_{D_i \in R(I)} \text{Rel}(D_i, I)$ where Rel(D, I) is the judged relevance, $R(I)$ is the set of all relevant documents for topic I. (This year, for some of the input judgments, {0, 1, 2} 3 levels of relevance is used.) Because judged irrelevant documents have relevance score 0, they will not have any effect in the model.

For the pseudo relevant documents, $D \in \mathrm{D_P}$, we have to use Bayes rule to expand the $P(D \mid I)$ term just as in (1), thus, arriving at the following decomposition,

$$\begin{aligned} P(r \mid I) &= \sum_{D_i \in \mathrm{D_T}} P(r \mid D_i) P(D_i \mid I) + \sum_{D_j \in \mathrm{D_P}} P(r \mid D_j) \frac{P(I \mid D_j) P(D_j)}{P(I)} \\ &= \sum_{\mathrm{D_T}} P(r \mid D_i) P(D_i \mid I) + \sum_{\mathrm{D_P}} P(r \mid D_j) P(I \mid D_j) P(D_j) \big/ P(I) \end{aligned} \qquad (3)$$

This final score $P(r \mid I)$ will be the weight for the expansion term r in the relevance model. The above derivations are exact, except there are some unknowns in the result: $P(I)$, $P(D_j)$, and the total collection-relevance $\sum_{D_i \in \mathrm{C}} \mathrm{Rel}(D_i, I)$.

Notice that these probabilities may depend on topic, and special care must be taken to deal with these values, as they directly affect the relative importance of true relevant documents w.r.t. pseudo relevant documents. We propose a justified and effective way of dealing with this balancing.

2.1 Balancing weights from pseudo- and true- relevant documents

There are several key problems to solve in order to properly balance the weights. *Firstly*, since some of the probabilities will never be known, we want to use a hyper-parameter to weight the two sources, and tune the weight on training data. *Secondly*, the numbers of (true and pseudo) feedback documents can be different across topics, thus, a larger $\mathrm{D_T}$ set would probably lead to a bias toward terms from the judged set and thus the optimal hyper parameter would vary. That's why a normalization step needs to happen so that the numbers of feedback documents – true v.s. pseudo – will not affect the relative weights of the terms from these two sources. *Thirdly*, note $P(I \mid D_j)$ the query generation probability will have large variations across topics, as a query can contain different number of query terms which can be popular or rare. Thus, we need to reduce this variance so that the hyper parameter is less variable across topics, and it's easier to learn an effective hyper parameter value which will be applied across all topics.

From equation (3), since we don't know the judgments of all the documents, we estimate $\mathrm{P}(D_i \mid I)$ on the given samples $\mathrm{D_T}$, this gives us the resulting term $\hat{P}(D_i \mid I) = \mathrm{Rel}(D_i, I) \big/ \sum_{D_i \in \mathrm{D_T}} \mathrm{Rel}(D_i, I)$, call it the empirical relevance distribution.

$$P(r \mid I) \hat{=} \sum_{\mathrm{D_T}} P(r \mid D_i) \hat{P}(D_i \mid I) + \sum_{\mathrm{D_P}} P(r \mid D_j) P(I \mid D_j) P(D_j) \big/ P(I) \qquad (4)$$

We denote the hyper parameter α, and parameterize $P(r \mid I)$ in α as follows,

$$P^{\alpha}(r \mid I) = \alpha \sum_{\mathrm{D_T}} P(r \mid D_i) \frac{\mathrm{Rel}(D_i, I)}{\sum_{\mathrm{D_T}} \mathrm{Rel}(D_i, I)} + (1-\alpha) \sum_{\mathrm{D_P}} P(r \mid D_j) P(D_j) \frac{P(I \mid D_j)}{P(I)} \qquad (5)$$

$P(D_j)$ is the document prior, if we assume uniform, it doesn't matter what exact value it takes, as tuning α will effectively cancel this constant term out. For an analogy with the empirical relevance distribution, we also use the empirical prior, $1/|\mathrm{D_P}|$. These two empirical distributions act as normalizers to the two sources, ensuring the weights are normalized properly with respect to the numbers of samples in both sources.

The above solves the *first* and *second* problems outlined at the beginning of the subsection. For the last problem $P(I)$, we don't know how exactly should it be determined, but we do know it should be dependent on the collection and on the topic, and also we expect it to provide proper balancing effect for the term $P(I \mid D_j)$, which varies a lot across topics. Given the requirements, we can have several alternative assignments for $P(I)$, **a)** $\max_{D \in \mathrm{C}}\{P(I \mid D)\}$, **b)** $\mathrm{avg}_{D_j \in \mathrm{TOPN}} \{P(I \mid D_j)\}$. Intuitively, if we neglect α, assignment (a) means we assume that a relevant document is as good as the best scoring document in the collection, while assignment (b) means it is as good as the average score of the top N results. We avoided using the simplest $P(I \mid C)$ – the query generation probability of the collection model – as an approximation for $P(I)$, because $P(I \mid D_j)$ would vary more wildly than $P(I \mid C)$ depending on how well the documents in the collection are matching with the topics, while $P(I \mid C)$ is ignorant of the documents.

$$P(I) = \max_{D \in \mathrm{C}} P(I \mid D) \qquad (6)$$

$$P(I) = \underset{D \in \mathrm{TOP}_N}{\mathrm{avg}} P(I \mid D) \qquad (7)$$

Without experiments, it is difficult to see how the two different assignments would behave because there is a regulating parameter α outside of the scope of the assignments that will determine how much weight to put on the two sources (true and pseudo relevance feedback). In general, we should select assignments that produce the most stable optimal regulating α across all topics, or more directly select the best assignment according to the final retrieval effectiveness measure (such as overall MAP etc.).

3 Experiments

3.1 Training and test datasets

We train all the parameters in the model on the training set by parameter sweeps. To create the training set, topics and assessments from the following track were used: TREC Terabyte 2004, 2005 and 2006 tracks and Million Query track 2007. We require each training topic to contain at least 40 judgments, and use 2/3 of the judgements as feedback documents and the rest 1/3 as evaluation data. For the feedback judgments, at least 4 should be relevant and 5 non relevant. Other topics are filtered out. We keep a ratio of 3:1 for MQ topics and Terabyte topics just to mimic the testing conditions and evaluate on the whole set using MAP. Total number of topics is 296, with 74 Terabyte topics. Although for training efficiency reasons we sampled a smaller subset of 113 training topics from this whole set.

The selection of the feedback documents is crucial for creating a proper training set. However, since we don't have the same results data from previous tracks as the relevance feedback track organizers do, we can only try to be as close to the test data as possible. Since the TREC judgment pool biases toward top ranked documents, although maybe not as high a bias as the feedback set for the test data, we still simply used a 2/3 random sample of the judgments from the pool as feedback and the rest as evaluation data. We fear that directly using top ranked results from the baseline run would bias the results too much, but we did not try, with the official test data now available we could try and report rankings of the feedback documents for the training sets that we may use, and compare to that of the test set.

3.2 Parameter tuning and performance analysis

Submitted results are shown in Table 1. All parameters were tuned on training set. Because of the large number of parameters to tune (especially conditioned on the different setups), we fix all the rest parameters and tune only one parameter at a time. We fix smoothing parameters first for the baseline Set A algorithm, and then keep the same parameters for feedback runs. When tuning feedback runs, we only vary the number of feedback terms and feedback documents, α and the weight for the original query.

From the results, we can see that the optimal α is quite stable across different feedback settings, meaning we have done proper balancing. Also, we did a more detailed per topic optimal α analysis on Set C and Set D, and results show that although the optimal α varies across topics, there are 1/3 of the topics where α can be 0-1 without changing performance; most of these topics are either too hard (MAP<0.05) or too easy (MAP>0.5). In 3/4 of the other topics where α makes a difference, optimal $\alpha > 0.5$, meaning true relevance feedback is emphasized more than the top ranking result. If we take the optimal MAP value for each topic, we get the upper bound for the performance we can get if we vary α, the upper bound on the training set turns out to be MAP =0.1957 for α in the set {0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} and with all

Table 1. Submited results with common parameters: dependency model initial queries, phrase smoothing Dirichlet with mu=4000, and term smoothing mu=1700, #feedback terms = 50, original query weight=0.7, max assignment for $P(I)$. Other parameters shown below:

Runs: Runtag	**Stop-words**	**#fb Docs**	α	**Training: MAP, RR, P@10**	**Official Test*: MAP RR P@10, mtc statAP**
CMURF08.A1	Included	10	<1	0.1738, 0.4374, 0.1858	0.1008, 0.4089, 0.2548 0.04025, 0.1646
CMURF08.B1	Excluded	10	0.7	0.1905, 0.4376, 0.1929	0.1034, 0.3918, 0.2452 0.04299, 0.1777
CMURF08.C1	Excluded	10	0.8	0.1878, 0.4323, 0.1956	0.1039, 0.4122, 0.2452 0.04310, 0.1817
CMURF08.D1	Excluded	10	0.8	0.1905, 0.4438, **0.1973**	0.0989, 0.3914, 0.2452 **0.04312**, 0.1799
CMURF08.E1	Excluded	10	0.8	0.1876, **0.4631**, 0.1938	**0.1094**, **0.4397**, **0.2774** 0.04184, 0.1817
CMURF08.B2	Included	10	0.8	0.1898, 0.4383, 0.1885	0.0985, 0.4157, 0.2387 0.04127, 0.1694
CMURF08.C2	Excluded	12	0.8	0.1883, 0.4367, 0.1947	0.1040, 0.4120, 0.2452 0.04313, **0.1820**
CMURF08.D2	Excluded	12	0.8	**0.1907**, 0.4481, 0.1965	0.0991, 0.3910, 0.2452 **0.04315**, 0.1804

***** Reported test metrics **MAP RR** and **P@10** are evaluated on the 31 Terabyte track topics, the **mtc** and **statAP** measures are evaluated on MQ topics, with 237 and 208 valid topics respectively.

other parameters set as runs CMURF08.C1 and D1, especially, the original query weight is set as 0.7, which restricts the effects of the expansion query.

When only with one (the top) relevant document for feedback, on the test set, our model (α = 0.7) performs significantly better than relevance feedback alone (α = 1), MAP 0.1034 v.s. 0.1005, with significance level $p < 0.004$ by a paired sign test. Although it is not significantly better than pseudo relevance feedback alone (α = 0, MAP=0.09918).

We also show per topic how our results compare to the median system in Figure 1. The results are worse than the median on average. And a comparison with other systems show that the baseline is much worse than other systems. However, comparing the relative gains in MAP by using more relevance data (Table 1), the constructed training set and the official test data have a similar trend, and the most performance gain is in the 10% range. On the training data, this gain is achieved right at set B, while on the test data, the best performance is at set E. This suggests that a randomly sampled feedback document (what's done on the training topics) is more informative than a top ranked relevant document (for the test case). Also, this relative gain is smaller than the best systems, and it might be because of the worse baseline. More experiments need to be done to re-establish a baseline and confirm.

4 Conclusions

We presented a coherent model of incorporating pseudo relevance feedback together with true relevance feedback. We showed that the difficulty lies mainly in balancing term weights coming from the two different sources. Several steps have been taken to deal with weight balancing, and reasonable alternative solutions exist. Results show that merging relevance feedback and pseudo relevance feedback is significantly better than relevance feedback alone when feedback documents are few. Randomly sampled relevant document is carrying more information than a top ranked relevant document, in terms of feedback performance.

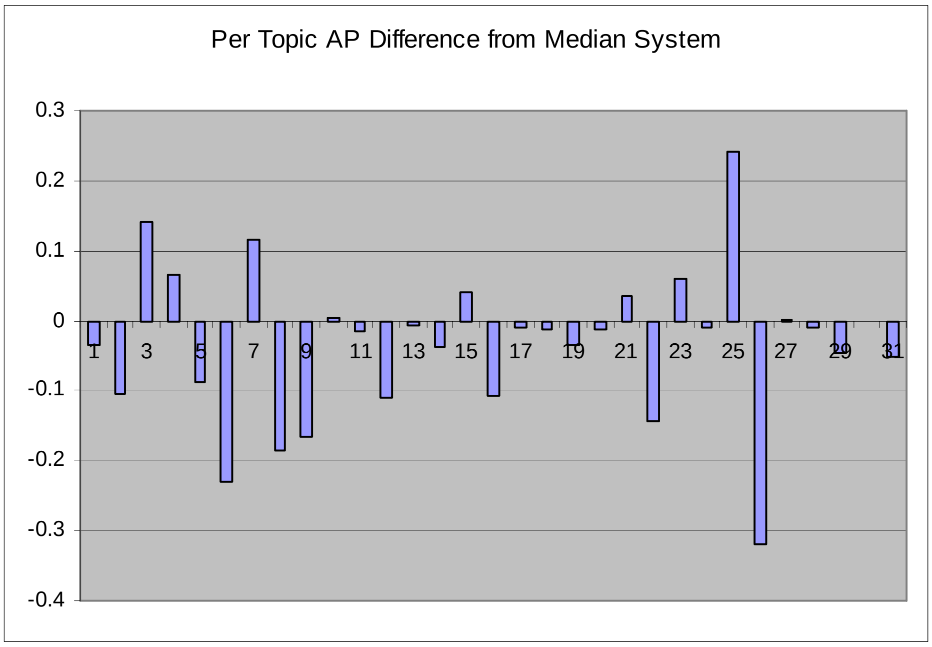

Figure 1. Comparing CMURF08.E1 with the median system.

For future work, we would like to explore ways of modeling negative feedback, more finely controlled term extraction, such as restricting a window around each query term, to get more precise feedback terms. We are also interested in investigating how perturbing the feedback documents would affect results.

References

Donald Metzler, Trevor Strohman, Yun Zhou and W.B. Croft, 2005. Indri at TREC 2005: Terabyte Track. In *Proceedings of the 14th Text REtrieval Conference (TREC)*.

Jingjing Li and Hongfei Yan, 2006. Peking University at the TREC 2006 Terabyte Track. In *Proceedings of the 15th Text REtrieval Conference (TREC)*.

J. J. Rochio, 1971. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System Experiments in Automatic Document Processing*, pages 313--323.

Xuanhui Wang, Hui Fang and Chengxiang Zhai, 2008. A Study of Methods for Negative Relevance Feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)*, pages 219--226.

Victor Lavrenko and W.B. Croft, 2001. Relevance-Based Language Models. *Proceedings of the 24th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (SIGIR '01)*, pages 120--127.

Yangbo Zhu, Le Zhao and Jamie Callan, 2007. Structured Queries for Legal Search. In *Proceedings of the 2007 Text REtrieval Conference (TREC 2007)*.