PRIS at 2009 Relevance Feedback track: Experiments in Language Model for Relevance Feedback

Si Li, Xinsheng Li, Hao Zhang, Sanyuan Gao, Guang Chen, Jun Guo
School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications
Beijing, P.R. China, 100876
ls198cf@gmail.com

Abstract:

This paper describes BUPT (pris) participation in Relevance Feedback Track 2009. The track has two phrases. In the first phrase, 5 documents are submitted based on the results of the k-means. In the second phrase, language model is used to relevance feedback for query expansion.

1 Introduction

This year, the Relevance Feedback Track has two tasks [1]. In the first phase, participants should determine up to 5 documents for each topic that they desire judged. In the second phase, the results of phase 1 will be used as judged document RF input for phase 2 of the track.

The PRIS-RF system is submitted by Pattern Recognition and Intelligent System Lab at Beijing University of Posts and Telecommunications. In the first phrase, clustering algorithm is employed to get the center documents. In the second phrase, relevance feedback algorithm is used. Our system adopts language model based on weekly semi-supervised machine learning for query expansion. According to only a bit of labeled documents, clustering algorithm and bootstrapping method are used. In the first stage, the 5 given documents are regarded as the center documents. The more documents can be labeled based on k-nearest neighbors (K-NN) clustering algorithm. At second stage, language model is used in the labeled documents for query expansion. Then based on bootstrapping method, the two staged are iterated until the relevance retrieval ranking list is stable. The basic ad-hoc retrieval platform is based on the Indri Retrieval Toolkit [2].

The remainder of this paper is organized as follows. In section 2, a briefly system overview is presented. Section 3 introduces the topic retrieval part. Section 4 describes the relevance feedback system. Evaluation results are shown in section 5.

2 System Overview

The framework of the PRIS-RF system is shown in Figure 1. The preprocessing part is designed to extract content of the permalink HTML pages, and some rules are set to process abbreviations. We only use the permalink HTML pages for retrieval. These HTML pages are parsed and texts are reserved. The hyper-links, scripts, style information in the web pages and all html tags are discarded. The topic retrieval part based on the Indri Retrieval Toolkit tries structured search on the document-level retrieval. Then we get the baseline of topic relevance ranking list. The feedback is carried out based on the baseline. Language model is the main model in feedback

algorithm.

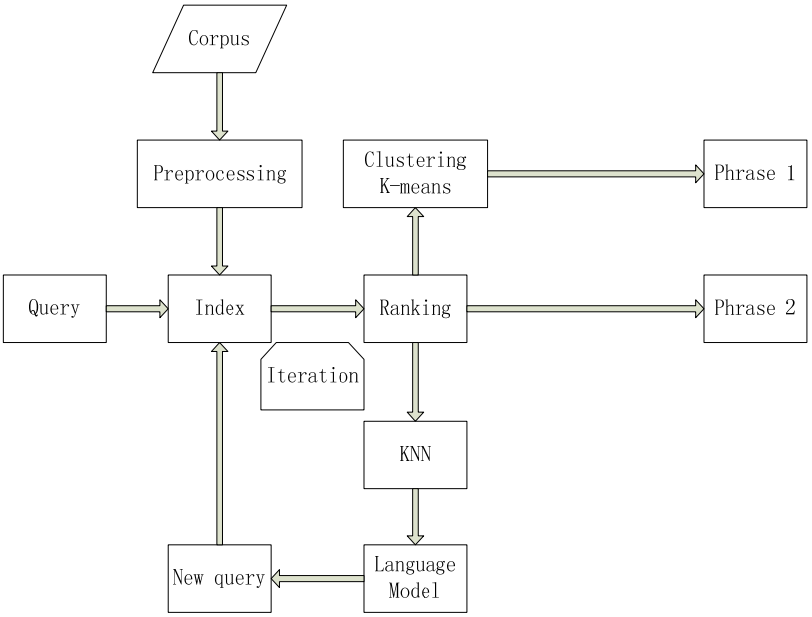

Figure 1 System Framework

3 Topic Relevance Retrieval

In this part, Indri is used to build index and search. Structured search is contained in the query language. The title field of the wed page is built into index. And some query languages are used in the baseline query. The following is an example of the baseline query.

```
<query>
    <number>18</number>
    <text>#5 (wedding budget calculator).(title) wedding budget calculator</text>
</query>
```

4 Language Model for Relevance Feedback

The main issues in relevance feedback are how to select relevant documents from the retrieved documents, and how to select expansion terms. Here we deal with the problem of selecting better expansion terms. The problem in traditional relevance feedback obtaining a set of expansion terms from the relevance retrieved documents that may have low precision. If a method can select better expansion terms from the relevance documents, it can almost certainly improve the effect of retrieval. The main process we have done is described in Algorithm 1.

Algorithm 1: The PRIS- RF algorithm

INPUT: 5 labeled documents

OUTPUT: a group of expansion terms

1 . Find the top k nearest neighbors to the 5 labeled documents to construct a labeled collection.

2 . Based on the Language Model, the expansion terms are extracted from the labeled collection which is got by the first step.

4.1 Clustering Model

5 labeled documents can be got for each query. But it's too few to get the expansion term for the topic. So clustering is employed to get more relevance documents from the retrieved documents based on those labeled documents. The clustering model follows the hypothesis that the documents which are near to the labeled documents have the same label as the labeled documents. So the k-nearest neighbors (K-NN) clustering method [3] is adopted to find the relevance documents. In the KNN, each labeled document plays a central role. Each document is represented by VSM. The similarity is calculated between the labeled and non-labeled documents in order to delete the duplicate documents and label the unlabeled documents. The top n documents are labeled and used to extract the expansion terms.

The top n documents cluster a relevance class and another non-relevance class. Because the 5 labeled documents contains relevance documents and non-relevance documents. Since some cross-documents appear in the relevance class and non-relevance class, there will be some noise generated. So the expansion terms extracted from those cross-documents must be inaccurate. To eliminate the noise, we have to remove the cross-documents from the relevance class and non-relevance class. After the eliminated process, we got the pure relevance class and non-relevance class.

4.2 The Language Model

The query is treated as a random event generated according to a probability distribution by the language model method to IR, developed by Ponte and Croft. Here, he made a simply assumption that the user of an IR system will have an idea of a prototypical document in which he or she is interested and will choose query terms likely to occur in documents similar to that prototype. Viewed this way, one can then estimate a model of the term generation probabilities for the query terms for each document. And then he can rank the documents according to the probability of generating the query. This language model is our improved language model's foundation. In his language modeling approach the probability of generating the query terms for each document can be generate. So our improved language model change the query terms for each document to each terms in the relevance class and non-relevance class which generated on the cluster model. Then we can get the probability of each term in those two classes. Each term is ranked by the probability .We extract the top n terms as the expansion terms. The detail instruction is as follows.

In the language model approach to IR, each document and each term of documents are ranked according to the estimate of producing the terms according to the language model. So to get the probability of terms generation is the first step. The terms generation probability $p(Q|M_d)$, is the probability of producing the terms given the language model of document *d*. This probability will be estimated starting with the maximum likelihood estimate of the probability of term *t* in document *d*:

$$\hat{P}_{ml}(t \mid M_d) = \frac{tf_{(t,d)}}{dl_d} \qquad (1)$$

$tf_{(t,d)}$ is the raw term frequency of term *t* in document *d* and dl_d is the total number of tokens in document *d*. A simplifying assumption will be made. Assume that given a particular language model, the terms occur independently. Based on the assumption, the maximum likelihood estimator gives rise to the ranking formula $\prod_{t \in D} \hat{P}_{ml}(t|M_d)$ for each document.

But there is an insufficient data problem [4] for the reliable estimation of maximum likelihood. The insufficient data problem is that some documents miss one or more of the query terms. But we do not wish to give a probability of zero for those documents. Because doing so, a document missing even one of the queries would not be retrieved. To solve the problem of insufficient data, we need an estimate of a larger amount of data. That estimate is the mean probability estimate of *t* in documents containing it:

$$\hat{P}_{avg}(t) = \frac{\sum_{d(t \in d)} \hat{P}_{ml}(t \mid M_d)}{df_t} \tag{2}$$

df_t is the document frequency of *t*. This is a more robust statistic in the sense that we have a lot more data from which to estimate it, but another problem appears. Each document containing *t* drawn from the same language model cannot be assumed, and so some risk is contained in using the mean to estimate $p(t|M_d)$. Furthermore, if we use the average number of the probability of the term, the distinction between documents with different term frequencies will be ignored. In order to minimize the risk, the mean will be used to moderate the maximum likelihood estimator by combining the two estimates using the geometric distribution as follows:

$$\hat{R}_{t,d} = \left(\frac{1.0}{(1.0 + \overline{f_t})}\right) \times \left(\frac{\overline{f_t}}{(1.0 + \overline{f_t})}\right)^{tf_{t,d}} \tag{3}$$

f_t is the mean term frequency of term *t* in documents where *t* occurs normalized by document length.

Using the geometric distribution has several reasons. In the first place, the mean of the distribution is equal to $\bar{f}_t$ which is the mean probability of occurrence. Secondly, the variance of this distribution is larger than the mean. Finally, this function is defined in terms of only the mean and the *tf* so it can be computed without adding to the space overhead of the index and in minimal time.

So the estimate of the probability of producing the query for a given document model as follows:

$$\hat{P}(D \mid M_d) = \prod_{t \in D} \begin{cases} P_{ml}(t,d)^{(1.0-\hat{R}_{t,d})} \times P_{avg}(t)^{\hat{R}_{t,d}} & if, tf_{(t,d)} > 0 \\ \dfrac{cf_t}{cs} & otherwise \end{cases} \tag{4}$$

cf_t is the count of term *t* in the relevance class and non-relevance class . *cs* is the total number of tokens in those two classes. This function is computed for the probability of each terms in the documents are ranked accordingly.

4.3 Iteration

After this selection, the top n terms of the ranked list are chosen as final expansion terms. The final expanded query is combined with the original query using linear interpolation, weighted by parameter λ. Then we repeat the algorithm 1. This process constructs an iteration process. The combining parameterλ is set to be 0.8 firstly. Then this parameter decrease to 0.2 with the 0.2 decrement rate as the time of iteration increasing. When the top 2000 documents in the retrieval result do not change, we think the iteration process becomes stable. All process finished.

5 Submission and Evaluation Results

There are 8 results we submitted, one is the PRIS-RF baseline, and the other 7 results are got from feedback relevance model based on phrase 1 results. The whole experiments we have done are based on category B. The score of the baseline is 0.4833. The official evaluation results of the submitted 7 runs are listed in the following tables.

Table 1 Relevance Feedback results

	PRIS.hit2.2	PRIS.ilps.1	PRIS.PRIS.1	PRIS.Sab.1	PRIS.SIEL.1	PRIS.twen.1	PRIS.UCSC.1
emap	0.0328633	0.0330224	**0.0352531**	0.0325163	0.0313367	0.0307429	0.0325286
stAP	0.161541	0.172208	0.153682	**0.178729**	0.147595	0.139995	0.133959

References

[1] http://groups.google.com/group/trec-relfeed/topics?hl=en

[2] http://www.lemurproject.org/indri/.

[3] Kyung Song Lee, W. ,James Allan. 2008. A Cluster-Based Resampling Method for Pseudo-Relevance Feedback. The 31st annual international ACM SIGIR conference on Research and development in information retrieval .pp.235-242.

[4] W.Bruce Croft.2002. Advances in Information Retrieval. pp.74-172