DUTIR at TREC 2008 Relevance Feedback Track

Xiaoling Sun, Yuan Lin, Hongfei Lin, Zhihao Yang
Department of Computer Science and Technology, Dalian University of Technology
No 2 LingGong Road Shahekou District, Dalian 116023, China.
sunxiaoling1234@126.com, ly830310@163.com, {hflin,yangzh}@dlut.edu.cn

Abstract

This paper details our experiments carried out at TREC 2008 Relevance Feedback Track. We focused on the analysis of feedback documents, both relevant and non-relevant, to explore more useful information to improve retrieval performance. In our experiments, local co-occurrence model and a Rocchio formula were used to select good expansion terms. Five runs were submitted. These runs used different amount of relevance info for analysis.

1、 Introduction

Relevance feedback, which traditionally uses the terms in the relevance documents to enrich the user's initial query, is an effective method for improving retrieval performance. However, there has been comparatively few research advances in RF in recent years. There is no general agreement of what the best RF approach is, or what relative benefits and costs of the various approaches are [1].

Setting up a framework to look at these separate effects for future research will be an important goal for this track. It is anticipated that this will be at least a 2 year effort. The first year will investigate aspects of relevance feedback that can be looked at without user involvement and will use one round of batch relevance judgments. There are plenty of issues to look at even with this simple setup. The second year will start to loosen this up and examine more realistic decision procedures for determination of documents to be looked at (e.g., arbitrary rounds of judgments based on results of previous judgments.) User interfaces will not be looked at during the first two years.

Indri search engine which combines the language modeling and inference network approaches were used in our experiments. Many studies have found that the terms that co-occur with the query terms frequently are often related to the query [2]. So our algorithm is based on local co-occurrence method, and uses a Rocchio formula to filter out bad expansion terms.

The other sections are organized as follow: Section 2 describes the collection used in the RF track and preprocess. Section 3 is our index environment. Section 4 explains the methods and official runs. Section 5 details the results and discussion on them. Section 6 draws some conclusions form the work.

2、 Collection and Preprocess

The Terabyte document collection is used for RF track. The GOV2 corpus, which contains a large proportion of the pages in .gov domain, is used as the collection. It is made up of about 25 million documents comprising about 426GB of document source. Topics are a subset of those used for either the 2007 Million Query track or the 2005, 2006 Terabyte tracks. In particular, the Million Query track topics furnish a source of ranked judgments that are system independent (i.e., ordered by the algorithms that chose which documents to judge).

The html format pages are preprocessed in the corpus, and the others are unchanged. The basic idea of relevance feedback is to extract expansion terms from the relevance documents to formulate a new query for a second round retrieval [3]. If the words added to the original query are unrelated to the topic, the quality of the retrieval is likely to be degraded. Since the web is a highly volatile and heterogeneous information source and usually contains various types of materials that are not related to the topic of the web-page, such as navigation, decoration, interaction and the others. These are considered noises and harmful to retrieval performance. Therefore, it is necessary to filter out the noisy information.

DOM based method is used in our experiment. First, the page is passed through an HTML parser that creates a DOM tree representation of the web page. We use Htmlparser as our HTML parser, which takes care of correcting the HTML, therefore we do not have to deal with error resiliency. Once processed, the resulting DOM document can be seamlessly shown as a webpage to the end-user as if it was HTML. The DOM tree is hierarchically arranged and can be analyzed in sections or as a whole. We navigate the DOM tree recursively, using a series of different filtering techniques to remove and modify specific nodes and leave only the content behind. In the experiment, we just remove some specific nodes, such as <script>, <img>, <style> and so on, to filter out the noisy information. A further work can be done to extract the useful content of the webpage. Then, the tags of the html pages are removed. At last, all the pages in the corpus are converted to trectext format files.

In order to increase retrieval speed, we split the documents into 4 pieces, and build each index on a different machine. So we use 4 PCs for both indexing and retrieval all the GOV2 corpus documents. The PC has 2.19GHz CPU with 2GB RAM running Windows XP.

First, we index the whole GOV2 collection with no special document or link structure indexed. Second, we stem all documents by using the Porter stemmer. Third, we delete noise words using the stopword list provided by Indri. After all the documents are indexed, the size of full-text index is about 135GB.

3、 Methods and Official Runs

3.1 Local co-occurrence model

Xu and Croft’s experiment had showed that local context analysis, using the co-occurrence information between terms, can perform very well [4] on query expansion. So, in our experiment, we use local co-occurrence method to select good terms. The sentence-level window is set for the co-occurrence of query and candidate terms. In the past, document-level window was often used for statistic co-occurrence information, but the size might be too rough to reflect the degree of correlation between two terms. In addition, when the document is very long, it maybe involves many various themes, if the size of window is large, “Theme offsets” problem will occur. That is why we select such a small size.

The method is similar to [5], but the size of window is a sentence. The co-occurrence between term t and query term q is defined as follows:

$$Cooc(t,q\,|\,S)=\min(1,tf\,(t\,|\,S)*tf\,(q\,|\,S)) \tag{1}$$

If t and q occurrence in a sentence, no matter how many times, we define it as 1. Then the degree of co-occurrence between t and q in the relevance document sets RE is:

$$Cood(t,q \mid RE) = \frac{\sum_{D \in RE} \sum_{S \in D} Cooc(t,q \mid S)}{n} \tag{2}$$

At last, the cohesion degree between term t and query Q is:

$$Cohd(t,Q \mid RE) = \prod_{q \in Q} Cood(t,q \mid RE) \tag{3}$$

Then the top K terms can be selected as expansion terms.

3.2 A Rocchio formula

The co-occurrence model only considers the relevance documents, but non-relevance documents also help to improve the retrieval accuracy by lowering the rank of the documents similar to the non-relevant feedback documents. The expansion terms should make the largest difference between the relevance feedback documents and the non-relevance documents. So the non-relevance documents should be involved to select good expansion terms.

The term weight is assigned using the Rocchio formula applied to INQUERY's version 2.1 weighting scheme [6]. The top-K non-query terms in the following order are chosen and weighted using a Rocchio formula, β=0.75， γ=0.25:

$$Weight(t) = \beta \cdot \frac{1}{n_r} \sum_{rel} belief - \gamma \cdot \frac{1}{n_{nr}} \sum_{nonrel} belief \tag{4}$$

And the belief for term t in doc d is calculated by the formula:

$$belief(t,d) = 0.4 + 0.6 \cdot (0.4 \cdot \min(1, \frac{200}{\max tf_d}) + 0.6 \cdot \frac{\log(tf_{t,d} + 0.5)}{\log(\max tf_d + 1)}) \cdot \frac{\log(n_t + 0.5 / N)}{\log(N+1)} \tag{5}$$

3.3 Official runs

Because of the different advantages of the two methods mentioned above, we merge the two methods. The candidate terms are re-ranked by the scores obtained by the two methods, and then the top-K terms are selected.

We remove the stop words using a stop words list provided by Indri Search Engine from the original query, and combine the residual words by Indri operator #combine.

In the other runs, we also remove stop words. The original query and expansion terms are combined into a new query of the form:

#combine (0.8 original-query 0.2 expansion terms)

We submitted 5 runs for Relevance Feedback Track:

A: Our baseline, we used the Indri retrieval system. This run is only a simple query likelihood run. We don't make any change to the topics, just remove the stop words.

B: 1 rel doc. We use the local co-occurrence model in the relevance documents. The window of co-occurrence is a sentence. Top 20 terms are selected to extend the original topics. The proportion of original topic terms to extended terms is 0.8:0.2; the other runs are the same proportion.

C: 3 rel docs and 3 nonrel docs. The two methods are used to select good expansion terms, because the non-relevant documents are available. 30 terms are selected.

D: 10 judged docs (superset of C, so at least 3 rel and 3 nonrel docs). The same to C methods.

E: Large amounts of judged docs (40 to 800). The same to C methods, but number of the expansion terms is larger. We select 70 terms because more feedback information can be used. This run examines the amount of improvement possible with more relevance info.

4、 Results

All 5 runs are summarized in table1.　Table 1 is based on judging a pool of the top 10 ranked documents from each run for a subset of 31 of the terabyte-track topics. The results of MTC and statAP algorithm are not listed here.

Run	P@5	P@10	MAP	R-Precision	bpref
DUTIRRF08.A1	0.2387	0.2290	0.1326	0.1633	0.2033
DUTIRRF08.B1	0.2774	0.2774	0.1523	0.1825	0.2213
DUTIRRF08.C1	0.3226	0.3000	0.1613	0.1956	0.2283
DUTIRRF08.D1	0.2968	0.2903	0.1675	0.2056	0.2349
DUTIRRF08.E1	0.3290	0.3258	0.1664	0.1890	0.2376

Table 1 Run submitted for top 10 ranked documents for a subset of 31 of the terabyte-track topics

From the results we can see, with the feedback information increases, the results also increase at the same time. DUTIRRF08.D1 is a little better than the other runs. But the runs performed not very well on the whole. The reasons maybe include: the simple algorithm, the number and quality of expansion terms and so on. DUTIRRF08.E1 uses more information, but the result is not improved. Therefore, good performance can be obtained “on the cheap”, by using just a relatively low number of positive judgments. When the number of expansion terms is increased, the terms selected are either unrelated to the query or is harmful, instead of helpful, to retrieval effectiveness.

5、 Conclusion

In the RF track, we use local co-occurrence model and a Rocchhio formula to select good expansion terms, but the results are not very well. The reason maybe, the sizes of the optimal query of different topics are different. It means that we cannot use the same query size for all the topics. In the future, we are interested in exploring the best query size and using the best methods for each different topic. In addition, expansion-based feedback maybe creates an inconsistent interpretation of the original and the expanded query, model-based feedback should be mainly considered.

References

[1] Proposal for a TREC 2008 Relevance Feedback Track

[2] Guihong Cao, Jian-Yun Nie, Jianfeng Gao and Stephen Robertson. Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In Proceedings of SIGIR ’2008, pp.243-250,2008.

[3] Shipeng Yu, Deng Cai, Ji-Rong Wen and Wei-Ying Ma. Improving Pseudo-Relevance Feedback in Web

Information Retrieval Using Web Page Segmentation. In Proceeding of WWW 2003.

[4] Xu J. X. and Croft W. B. Improving the Effectiveness of Information Retrieval with Local Context Analysis [J]. ACM Transaction on Information Systems, 2000, 18(1): 79-112.

[5] Ding Guodong, Bai Shuo and Wang Bin. Local Co-occurrence Based Query Expansion for Information Retrieval. Journal of Chinese Information Processing, 2006, 20(3): 84-91.

[6] James Allan, Lisa Ballesteros, James P. Callan, W. Bruce Croft and Zhihong Lu. Recent Experiment with INQUERY. In Fourth Text REtrieval Conference (TREC-4), 1995.