Relevance Feedback based on Constrained Clustering: FDU at TREC 09

Bingqing Wang and Xuanjing Huang

School of Computer Science and Technology
Fudan University
wbq@fudan.edu.cn

Abstract. We introduce our participation of the TREC Relevance Feedback(RF) TRACK in 2009. The RF09 TRACK is focused on the explicit relevant feedback, where a few relevant and irrelevant documents are available to each query. Our system is implemented under the framework of probabilistic language model. We apply the constrained clustering on the top returned documents and extract the expanded words to reform the query. We also extract the named entities from the explicit relevant documents to expand the query. The experiment was conducted on the ClueWeb09 TREC Category B, which is a new and huge test collection for the TREC TRACKs. The evaluation result shows the performance of the constrained clustering.

1 Introduction

Relevance Feedback(RF) utilizes the relevance evaluation information to reform the user's query so that the overall retrieval performance (such as MAP, NDCG) can be improved. Although it has been proposed for many years, some questions still remain unexplored. Therefore, TREC organizes the Relevance Feedback Track to offer a chance for researchers to study the problems in RF. In 2009, it is the second year of the RF Track and this year's task focuses on the explicit relevance feedback. The participants are given the queries together with a few relevant documents and irrelevant documents to reform the original query. The "ClueWeb09" test collection is the new test bed for the RF Track, which is a challenge for the researchers to handle huge amount of test collection. The detailed process of the RF Track can be found in the google's group[1]. The description of the ClueWeb 09 collection can be found on the website[2].

There are 50 test queries to be retrieved. The whole process of RF Track is composed of two phases. The first phase requires each participant to submit one or two runs. "Phase 1 run" returns 5 documents for each query. The relevance of the 5 documents will be evaluated and a small query relevance pool will be built by these small runs. The Phase 2 is based on the phase 1. In the phase 2, each participant is given their own and other team's evaluated "Phase 1 run". The word "evaluated" means that we can definitely know which documents of the "Phase 1" run are relevant and which are not.

According to the requirements above, we mainly use two techniques in the RF09 task.

[1] http://groups.google.com/group/trec-relfeed
[2] http://boston.lti.cs.cmu.edu/Data/clueweb09/

1 We conduct the constrained clustering on the top retrieved documents. The expanded words are extracted from the clustered pseudo relevant document set.
2 We extract the named entities from the relevant documents in the "Phase 1 run" and add these named entities as expanded terms to reform the query.

Constraint Clustering: the main character of the RF09 task is the explicit relevance feedback. However the few relevant documents are far from sufficient. We want to find more relevant documents by using the Phase 1 run. So it is intuitive to use constrained clustering on the top documents returned by the first-pass retrieval.

Named Entity: Considering that many queries are related with some events or famous persons and organizations. And these named entities usually show a low raw frequency compared with other single words in the documents. We empirically extract the name entities (Name, Location, Organization) from the relevant documents to expand the original query.

Since the new test bed is far bigger than previous TREC test collections, we conducted the experiments with the help of lemur/indir toolkit version 4.10[3], which provides us convenient utilities to process the TREC data set. The experiments are implemented with the indri toolkit, which is based on the probabilistic language model. The result shows the effectiveness of the proposed approach, which is not sensitive to input explicit relevance feedback information. While the effect of the named entity extraction depends on the quality of the Phase 1 run. The named entities can also be a good source for query expansion but need additional exploration.

The rest of this paper is structured as follows. Section 2 introduces the constrained clustering algorithm. Section 3 introduces the named entity extraction for query expansion. Section 4 describes our experiment result and discussion. The paper is concluded in Section 5.

2 Clustering for Query Expansion

2.1 Constrained Clustering

Constrained Clustering or semi-supervised clustering[1] is suitable for handling the explicit relevance feedback in our task, which is to conduct the clustering with some constraints on the instances. There are two types of constraints, the *must-link* constraint $c_=(x, y)$ and the *can-not-link* constraint $c_\neq(x, y)$, meaning that the instance x and y can or cannot be partitioned into the same cluster. The must-link constraint is transitive, meaning that $c_=(x, y), c_=(y, z) \Rightarrow c_=(x, z)$. The can-not-link constraint can be entailed.

Suppose that CC_i and CC_j are connected components and let $x \in CC_i$ and $y \in CC_j$.

- must-link is transitive: if $c_=(x, y)$ exits, then $\forall a \in CC_i$ and $b \in CC_j$, $c_=(a, b)$ holds
- can-not-link is entailed: if $c_\neq(x, y)$, then $\forall a \in CC_i$ and $b \in CC_j$, $c_\neq(a, b)$ holds

According to [1][3], given the input instances $\{x_1, x_2, \cdots, x_N\}$, where $x_i \in R^D$. The goal is to partition these instances into K cluster. We note the clusters as $\pi_k, k = 1, \cdots, K$, where $K = 2$ in our task. We use the indicator variable r_{nk} to represent the

[3] http://www.lemurproject.org

partition. r_{nk} takes the 1-of-K schemes, which means that $\sum_k r_{nk} = 1, r_{nk} \in \{0, 1\}$. If x_n belongs to the kth cluster $r_{nk} = 1$, and in other cases, $r_{nk} = 0$. Suppose that μ_k is the center of the kth cluster. The optimization criteria function is shown as follows,

$$Loss = \sum_{x_n \in \pi_k, n=1}^{N} (x_n - \mu_k)^2 = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk}(x_n - \mu_k)^2 \quad . \tag{1}$$

The algorithm to iteratively compute r_{nk} and μ_k is the variation of the K-Means algorithm, which is shown in Fig. 1. Details of the algorithm can be referred to (Wagstaff et al. 2001)[1][2]

Algorithm 1 COP-KMeans Algorithm

Input: N instances $\{x_1, x_2, \cdots, x_N\}$, $C_=$: set of pairwise must-link constraints, $C_{\neq}$: set of pairwise can-not-link constraints
Output: k clusters of the instances
Start
    Compute the transitive closure of the set $C_=$
    Replace all the instances in $C_=$ by a single instance with weight $|C_=|$
    Randomly generate cluster centers, $\mu_1, \mu_2, \cdots, \mu_k$
    repeat
        for $i = 1$ **do** N
            (1) assign x_i to nearest cluster ▷ nearest cluster center
            (2) if assignment of x_i always violates a constraint, then exit with failure
        end for
        Recalculate the cluster centers $\mu_1, \cdots, \mu_k$, take the weight of instance into count
    until *Loss* in Equ. 1 converges
End

2.2 Implementation in Document Cluster

We first made the initial retrieval for each query. The top 100 retrieved documents were collected, on which we would make clustering. The Phase 1 evaluated run was used to build the "Must-Link" and "Can-Not-Link" constraint. Then we used the COP-KMeans algorithm to cluster the top 100 documents. The cluster containing the relevant document was taken as the relevant document set R, while the other cluster was taken as the irrelevant documents U. We extracted the expanded words from R.

The divergence between two documents is important when we made clustering, which can be modeled by different approaches. We measured the divergence between documents in the vector space model, with each document d_i converted into a vector x_i. Each element of x_i is corresponded to a word in the document d_i, which is the BM25 term weight in the document. Considering the efficiency of the algorithm, we empirically discarded those words with a low BM25 weight. We believe that the remaining words could represent the main semantic content of the document.

For some queries, the Phase 1 run contains only the relevant documents(easy query) or only the irrelevant documents(difficult query). When no relevant documents were available, we assumed that the top 5 returned documents excluding the irrelevant ones were taken as the relevant documents R. When no irrelevant documents were available, we empirically selected 5 low-ranked documents as the irrelevant ones.

After clustering on the documents, we extracted the expanded words from the relevant document cluster. The expanded words were ranked by the sum of the BM25 weight over the relevant documents. The top expanded words were incorporated into the reformed indri query.

3 Named Entity

Previous work on the query has studied the classification of the query, which shown that some of the queries are entities focusing on some specific topics or domains. Named Entity (including People's Name, Location, Organization Names) are helpful to discriminate the relevant documents from the irrelevant documents. For example, given the query "out space universe", the organization name "NASA" is a good candidate expanded word.

In the conversional method, we select those words that show significant statistical connection the expanded terms. However, mixed with the high-frequency single words, the low-frequency named entities can only get a low rank the whole expansion terms. So we have to extract the named entities independently and add these entities into the indri query.

We extracted the named entities from the relevant documents in the Phase 1 run. We used the Stanford Named Entity Recognizer[5][4] to extract the named entities. with the default model in the package, trained on the CoNLL, MUC-6, MUC-7 and ACE named entity corpora.

We extracted the People's Name, Location and Organization Names and ranked these entities based on the raw frequency in the relevant documents. Some queries do not have relevant documents, so we did not extract named entities for these queries.

4 Experiment and Result

4.1 Overview of the Experiment Settings

We used the ClueWeb 09 TREC Category B test collection for both the Phase 1 and Phase 2 runs, which is a sub collection of the whole ClueWeb09 DataSet. The dataset was collected in January and February of year 2009. Statistics about the dataset is given in Tab. 1.

We conducted the experiment with the help of the Indri toolkit. We used the probabilistic language model as the retrieval model with the Dirichlet Smoothing method. In the Phase 1 run, the parameter μ in the Smoothing method is set to be the default value of 2500 in indri, which showed a poor experiment result. Since the language model is sensitive to the smoothing parameters, we set $\mu = 800$ in all the Phase 2 runs.

[4] http://nlp.stanford.edu/software/CRF-NER.shtml

Table 1. ClueWeb09 Category A and Category B, "Pages" shows the web pages count, "M" means Million, "Size" gives the compressed and uncompressed size of the collection, "Lang" is the language contained in the collection

ClueWeb09	**Pages**	**Size**	**Lang**
Category A	1,040M	5T / 25T	Many
Category B	50M	250G / 1T	English

In the Phase 2, we first made the baseline run according to the settings described above. The top 100 documents were extracted for clustering. The evaluated Phase 1 run were used to construct the constraints. After the constrained clustering, the top 20 expanded words were added to the original query. Meanwhile, the top 10 name entities extracted from the relevant document were also extracted and added to the original query.

For the Indri query language, the expanded query takes the form as *#weight(1.0 #combine(<query>) 1.0 #uw(<query>) 1.0 #combine(WordCluster) 1.0 #combine(NameEntity))*. Each term in the indri query could be given a weight. <query> is the original input query. We did not focused on the weight assignment, so we simply set all the weight to be 1.0 in our experiments.

4.2 Submissions and Evaluation Results

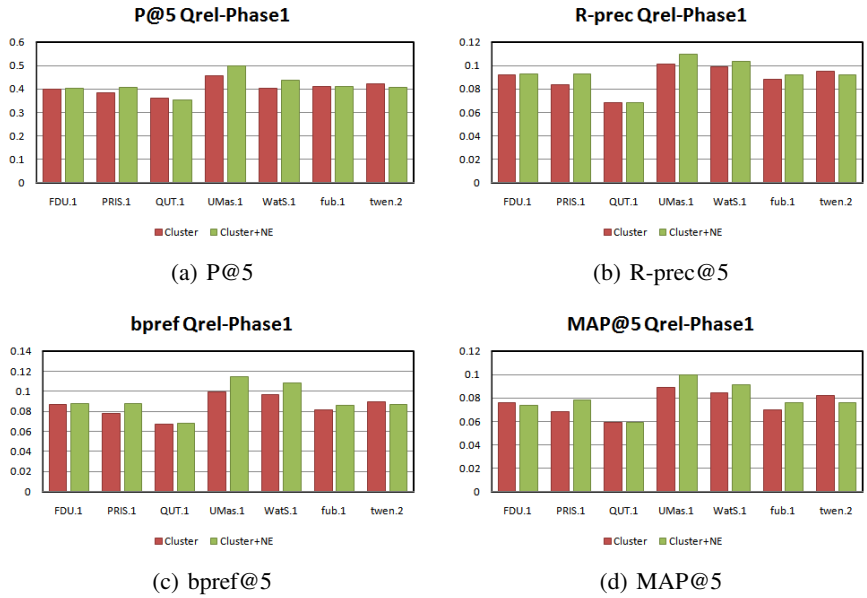
Table 2. Evaluation on Phase 2 runs

Submission	MAP@5	EMAP	StAP	Impt_MAP@5	Impt_StAP
Baseline	0.0501	NA	0.1645	NA	NA
FDU.1	0.0735	0.0439	0.2268	46.7%	37.9%
PRIS.1	0.0782	0.0440	0.2311	56.1%	40.5%
QUT.1	0.0594	0.0486	0.2386	18.6%	45.0%
UMas.1	0.0998	0.0461	0.2437	99.2%	48.1%
WatS.1	0.0915	0.0466	0.2382	82.6%	44.8%
fub.1	0.0761	0.0498	0.2450	51.9%	48.9%
twen.2	0.0763	0.0467	0.2285	52.3%	38.9%

We report the evaluation results in Tab. 2. The Phase 1 runs only retrieve 5 documents for each query. The Qrel-Phase1 is a small query relevance pool generated from all the Phase 1 submissions. We did not receive the qrels pool of Phase 2, because we only used the ClueWeb09 Category B.

In the Phase 2, the input is the top 100 documents of "Baseline" and the Phase 1 run, the output is the final submission which returns the top 2500 documents for each query. The evaluation result is the Million Query Style(EMAP and StAP). The MAP@5 evaluation is made by ourselves. For each Phase 2 final submissions, we evaluated the MAP of the top 5 documents by using the Qrel-Phase 1 so that we can compare the retrieval

performance before and after the relevance feedback. All the MAP@5 is improved significantly compared with the "Baseline", also higher than their corresponding "Phase 1 run". To compare the input "Phase 1 run" and the output "Phase 2 run", we use the Qrel-Phase 1 pool to evaluate them and present the detailed results in Fig. 1.



(a) P@5 (b) R-prec@5

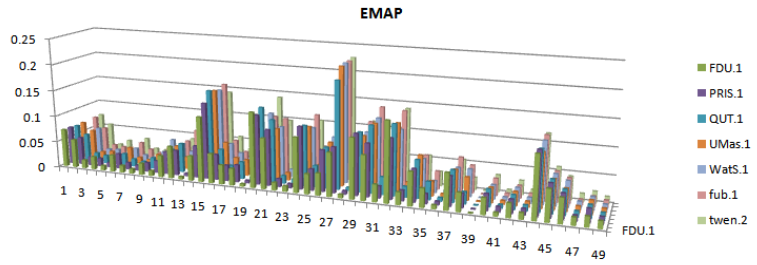
(c) bpref@5 (d) MAP@5

Fig. 1. Evaluation by Qrel-Phase 1

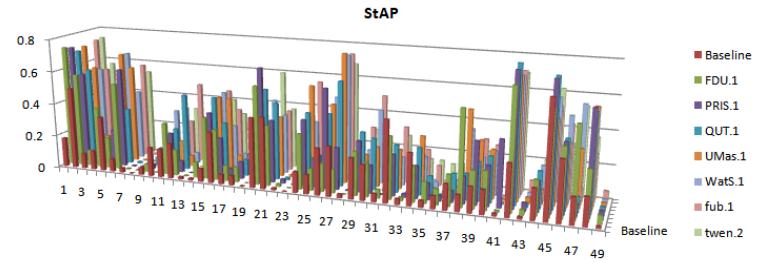
In Fig. 1, the red bars correspond to the Phase 2 run using the clustering method, the green bars corresponds to the Phase 2 run with both the clustering and the named entity extraction approach. We show the P@5, R-prec, bpref and MAP evaluation results. The results indicate interesting phenomena here. For most of the runs, query expansion can raise the precision and recall both. But for some good Phase 1 runs such as UMas.1 and WatS.1, query expansion will raise the recall significantly and sacrifice a little precision to achieve a better MAP evaluation. The named entity extraction can raise the overall performance but the improvement is marginal for some Phase 1 run.

Besides the results above, we also show the performance of each query here. The evaluation of the Phase 2 runs are the million query style. The Expected Mean Average Precision(EMAP)[6] and Statistical Sampling Precision(StAP)[7] of each query topic are presented in the Fig. 2.

The problem of the pseudo-relevance feedback can be shown in Fig. 2. For easy queries, the relevance feedback seems to help the performance but it seems still not effective to help those difficult queries.



(a)



(b)

Fig. 2. Evaluation of the Phase 2

5 Conclusion

We introduce our query expansion approach on the explicit relevance feedback of TREC RF09 TRACK. We adopt the constrained clustering and named entity extraction in the task and verify the effectiveness of our approach in the experiments.

We are focused on a more general approach for the explicit relevant feedback. In the future work, we will continue to study the approach to improve the constrained clustering model and adapt the model for the query expansion task.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant NO. 60673038), in part by the Ph.D Programs Foundation of Ministry of Education of China (Grant NO. 200802460066), and in part by the Shanghai Committee of Science and Technology, China (Grant No. 08511500302).

References

1. Ian Davidson and Sugato Basu: A Survey of Clustering with Instance Level Constraints. ACM Transactions on Knowledge Discovery from Data, (2007)
2. Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schroedl: Constrained K-Means Clustering with Background Knowledge. In Proceedings of ICML. pp: 577-584. (2001)
3. Christopher M. Bishop: Pattern Recognition and Machine Learning. Singapore: Springer 2006, pp: 179-224, (2006)
4. In-Ho Kang, GilChang Kim: Query Type Classification of web document retrieval. In Proceedings of the 26th SIGIR, pp: 64-71, (2003)
5. Jenny Rose Finkel, Trond Grenager and Christopher Manning: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In Proceedings of the 43nd ACL, pp: 363-370, (2005)
6. James Allan, Ben Carterette, Javed A. Aslam, Virgil Pavlu, Blagovest Dachev, Evangelos Kanoulas: Million Query Track 2007 Overview. In Proceedings of the 16th TREC, (2007)
7. James Allan, Javed A. Aslam, Ben Carterette, Virgil Pavlu, Evangelos Kanoulas: Million Query Track 2008 Overview. In Proceedings of the 17th TREC, (2008)
8. Claudio Carpineto, Renato De Mori, Giovanni Romano and Brigitte Bigi: An Information-Theoretic Approach to Automatic Query Expansion. ACM Transactions on Information Systems, Vol. 19, No. 1, pp: 1–27, (2001)