DLDE at web track: ad-hoc task

Jie Chen, Zhendong Niu, Yulong Shi, Changmin Zhang, and Weiyin Li

{sonyfe25cp@gmail.com,zniu@bit.edu.cn,syl_bit@163.com,
zcm_xinxi@163.com,li_weiyin@qq.com}

Abstract. *In this note paper, we report our experiment method at ad-hoc task of Web Track 2013. The goal of this task is to return a rank of documents order by relevance from a collection of static web pages. Our group used meta search to help query expansion as the first step, and then retrieved with the expansion query to get the search results and rerank them.*

Keywords: Ad-hoc search, query expansion, meta search

1 Introduction

The ad-hoc task of web track is given some topics to search out the most relevant documents from a large number of static web pages, ClueWeb2012 [6], which comprises about 870 million web pages collected between February 10, 2012 and May 10, 2012. Topics are short and ambiguous which are similar to queries of tradition web search. It is hard to get satisfied search results based on keyword match method. This year we use the WebClue2012-B13 dataset, a subset of WebClue2012 with 50 million web pages. Firstly, some preprocessing work was done to the dataset in order to remove the spams and noise data. Secondly, in order to get more semantic meanings, meta search approach was used to get some pages relevant to the topic as seeds , and then the semantic words were computed as expansion. Thirdly, the search results from the treated data with the expansion query were reranked.

2 Data preprocessing

Since the data set is too big to be operated directly and efficiently, the non-relevant data and spam were removed before the final query step. In our experiments , the Indri [7] tools and waterloo spam [2] were used to build the raw index files. Second, all the relevant pages were got from the index with the topic as the query. These relevant pages are the basic data of our experiments. In the process of parsing the web page , we found there are many noises such as advertisements and copy rights in body part, so the Block web content parser [3] ,developed by our lab, was introduced into this project to extract the main content. Third, a new index file with Lucene [8] was built for subsequent experiments. The reason was that our group had developed a web search system based on Lucene package.

3 Query Search

The query search phase was divided into two parts: query expansion and reranking. Each origin topic was expanded to a sematic word set as new query to get search results and treat them as the final results after reranking.

3.1 Query Expansion

Meta search Google search and bing search were used as meta search resource. Each search engine returns the top 200 pages about the topic, and then the page extract technology [3] was used to get main content.

Expansion Strategy Query expansion is a commonly used method helping search system to understand the origin query words. In our experiment, the local analysis [1] method was used to get the expansion words. Origin expansion words list was got by calculating occurrence number of each word and remove the stop words. With the help of Standford Parser [9] , developed by Standford Natural Language Processing Group, all the words in addition to nouns were removed and the top 30 were got as final expansion words list.

We treat the synonym of origin topic word as the denominator to get the weight of each expansion word

$$w_i = \frac{TF_i}{TF_{max}} \tag{1}$$

The final query expansion formula is descripted :

$$Query_{expan} = Query_{origin} + \sum_{i=1}^{n} w_i * Expan_i \tag{2}$$

n is the count of optional expansion terms.

3.2 Re-ranking Model

We aimed to re-ranking the search results with learning to rank technology which is on the rise in recent years. Learning to rank is a class of methods using machine learning to solve information retrieval problems. It is an effective way to combine different data features for ranking. The public available dataset ,LETOR [4], was used to train the rank model. Since a lot of features of LETOR we cannot get, we droped those columns and then trained the ranking model. The SVMRank [5] algorithm was used in this task and five-folds cross validation was done. The output model was directly applied to our experiment.

4 Experiments

Submited only one run this year and it didn't perform well. We think there are two main reasons that caused this result. One reason was our ClueWeb-B-13 dataset is too small to obtain the valid search results in the data preprocessing step. The other was caused by features we used to train the ranking model .Not only the number of features was small, but also there were gaps between LETOR dataset and ClueWeb2012.We need to do some transfer learning to train the ranking model next time.

5 Conculion

Our approach was descripted in this paper. This year we used web content technology to remove the noise of web page and used the meta search and query expansion method to understand the topic. Since the dataset we had less than on tenth of the whole one, we will validate our ideas with the whole dataset in the next year.

6 Acknowledgements

Thank organizers of TREC and NIST. This work is supported by the National Natural Science Foundation of China (no. 61250010) and the 111 Project of Beijing Institute of Technology.

References

1. Imran, H. ; Sharan, A. A framework for automatic query expansion Web Information Systems and Mining, Springer, 2010, 386-393
2. Cormack, G. V.; Smucker, M. D. & Clarke, C. L. Efficient and effective spam filtering and re-ranking for large web datasets Information retrieval, Springer, 2011, 14, 441-465
3. Lin, S.; Chen, J. ; Niu, Z. Combining a segmentation-like approach and a density-based approach in content extraction Tsinghua Science and Technology, TUP, 2012, 17, 256-264
4. Liu, T.-Y.; Xu, J.; Qin, T.; Xiong, W.; Li, H. Letor: Benchmark dataset for research on learning to rank for information retrieval Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval, 2007, 3-10
5. Chapelle, O.; Keerthi, S. S. Efficient algorithms for ranking with SVMs Information Retrieval, Springer, 2010, 13, 201-215
6. Carnegie Mellon University. The ClueWeb2012 Dataset [EB.OL]. http://boston.lti.cs.cmu.edu/clueweb12/
7. The Lemur Project. The Indri search engine software. http://lemurproject.org/indri.php
8. The ApacheSoftware Foundation. The Lucene Search Library. http://lucene.apache.org/
9. The Standford Natural Language processing Group. The Standford Parser. http://nlp.stanford.edu/software/lex-parser.shtml