DLDE at web track: ad-hoc task

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

No Institute Given

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 ranking of the documents order by relevance from the collection of static web pages. Our group use meta search to help query expansion as the first step , then do text retrieval 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 based on some given topics to search out the most relevant documents from a large number of static web pages, ClueWeb2012 [5], which comprises about 870 million web pages, collected between February 10, 2012 and May 10, 2012. Topics are similar to queries of tradition web search, short and ambiguity. It's hard to get better search results based on keywords match method. This year we use the WebClue2012-B13 dataset , a subset of WebClue2012 with 50 million web pages.First, we do some prepare word to the dataset in order to remove the spams and noise data. Second, in order to get more semantic meanings, we use meta search approach to get some pages relevant to the topic as seeds , then we can compute the semantic words as expansion. Third, we rerank the search results from the treated data with the expansion query.

2 Data preprocessing

Since the data set is too large to operate directly and efficiently, we remove the non-relevant data and spam before doing the final query step. In our experiments , we first use the Indri [6] tools and waterloo spam [**?**] to build the raw index files. Second, we get all the relevant pages 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 find here are many noises in body such as advertisements, copy rights, so the Block web content parser [2] , developed by our lab, is introduced into this project to extract the main content. Third, we build a new index file with Lucene [7] for subsequent experiments. The reason is that our group already have a web search system based on Lucene package.

3 Query Search

We divided the query search phase into two parts: query expansion and reranking. We expand each origin topic to be a sematic word set as new query to get search results and treate them as the final results after reranking.

3.1 Query Expansion

Meta search we use google search and bing search as a meta search resource. each search engine return the top 200 pages about the topic, and we use the page extract technology [2] to get main content.

Expansion Stractagcy Query expansion is a commonly used method help search system to understand the origin query words. In our experiment, we use the local analysis [1] method to get the expansion words. By calculating occurrence number of each word and remove the stop words to get the orgin expansion words list. With the help of Standford Parser [8] , developed by Standford Natural Language Processing Group, we remove all the words in addition to nouns and get the top 30 as the final expansion words list.

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

$$w_i = \frac{TF_i}{TF_m ax} \tag{1}$$

.We can get the final query expansion formula as below

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

3.2 Re-ranking Model

We aim to re-ranking the search results with learning to rank technology, on the rise in recent years. Learning to rank is a class of methods used machine learning to solove information retrieval problems. It is an effective way to combine different data features for ranking. We use the public data available for learning to rank, LETOR [3]. Since the LETOR has a lot of features but we can get less, we deleted those features of LETOR then train the ranking model. We use the SVMRank [4] algorithm in this task and do five-folds cross validation. The output model is directly applied to the experiment.

4 Experiments

This year we submit only one run and it performs bad. We think there are two main reasons for this result. One reason is our ClueWeb-B-13 dataset is too small to obtain the valid search result in the data preprocessing step. The other reason is features we used to train the ranking model .Not only the counts of features is a small number ,but there are gaps between LETOR dataset and ClueWeb2012.We need to use more features and suitable dataset to train the ranking model next time.

5 Conculion

We described our approach in this paper. This year we use web content technology to remove the noise of web page and use the meta search and query expansion method to understand the topic. But the final run performs bad. In the next year , we will use the whole dataset of ClueWeb2012 to verify our approach.

References

1. Imran, H. ; Sharan, A. A framework for automatic query expansion Web Information Systems and Mining, Springer, 2010, 386-393
2. 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
3. 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
4. Chapelle, O.; Keerthi, S. S. Efficient algorithms for ranking with SVMs Information Retrieval, Springer, 2010, 13, 201-215
5. Carnegie Mellon University. The ClueWeb2012 Dataset [EB.OL]. http://boston.lti.cs.cmu.edu/clueweb12/
6. The Lemur Project. The Indri search engine software. http://lemurproject.org/indri.php
7. The ApacheSoftware Foundation. The Lucene Search Library. http://lucene.apache.org/
8. The Standford Natural Language processing Group. The Standford Parser. http://nlp.stanford.edu/software/lex-parser.shtml