Peking University at the TREC 2006 Terabyte Track

Li Jingjing, Yan Hongfei
Networks and Distributed Systems Laboratory
School of Electronic Engineering and Computer Science
Peking University
Beijing,China,100871
{ljj,yhf}@net.pku.edu.cn

ABSTRACT

This paper details the experiments carried out at TREC 2006 Terabyte Track using Indri Search Engine. There were three tasks in the Terabyte track of TREC 2006, i.e. efficiency task, ad hoc task and named page finding task. We participated in two tasks, and submitted 5 runs for ad hoc task and 3 runs for named page task respectively. In ad hoc task, we looked at the importance of term proximity. In named page finding task, we cared more about the information of document structure and document prior.

Keywords

Evaluation, Search Engine, Terabyte.

1. INTRODUCTION

Indri is a new indexing and retrieval component developed by the University of Massachusetts for the Lemur Toolkit[2]. It is a scalable Search Engine that combines the language modeling and inference network approaches into a single framework, and it supports a robust query language, based on the INQUERY query language.

Our goals for this year's track were modest, to complete runs with Indri Search Engine, to gain a good understanding of Indri retrieval model, and to gain more experience in the procedure of evaluations. We have organized Chinese Web Track for several years[1]. In order to organize the Chinese Web Track better, the experience of TREC is important to us.

The remainder of the paper details the experimental work and results obtained for each of the two tasks we participated in. We describe the collection and tasks, indexing environment, retrieval techniques and details of the runs we submitted.

2. COLLECTION AND TASK SUMMARY

The GOV2 corpus, which contains a large proportion of the crawlable pages in .gov domain, was used as the collection. It is made up of about 25 million documents comprising about 426GB of document source.

The TREC 2006 Terabyte track consisted of three tasks. Ad hoc task was classical ad hoc retrieval task, a task which investigated the performance of systems searching on a static set of documents using a set of previously unseen topics. Participants were given 50 new topics, and for each of these topics participants were asked to return a ranked set of the 10,000 most relevant documents.

The efficiency task was a task whose aim was to provide a means for comparing efficiency and scalability issues in IR systems. We didn't participate in this task.

The named page finding task was to search a document by name. An effective retrieval system could return the page at or near rank one. Participants were given about 180 topics, each of which specified a document by name, and were asked to return top 1000 results for each of the topics.

3. INDEXING ENVIRONMENT

For various reasons, we ended up using a single PC for both indexing and retrieving all the GOV2 corpus documents. The PC had dual Xeon 2.8GHz CPUs with 4GB RAM running Red Hat Linux release 9.

We partitioned the GOV2 corpus into 9 roughly equal-sized pieces and built a separate index for each subset using Indri 4.2. All documents were stemmed using the Porter stemmer.

For the ad hoc task this year, we built index for all the documents, with no special document or link structure indexed. It took 1515 minutes to build the index and the size of full-text index was about 202GB.

For the named page finding task, we indexed title, mainbody, heading, and inlink fields, and it took 2236 minutes to build the index and the resulting index files was about 204GB in size.

4. RETRIEVAL TECHNIQUES

4.1 Ad hoc task

For ad hoc task, we adopted three techniques, i.e. query likelihood[3], dependence model[4] and pseudo-relevance feedback[5]. Query likelihood run was baseline run, which simply weighted each word equally. The dependence model is a mechanism for modeling term proximity features. Pseudo-relevance feedback, whose basic idea is to extract expansion terms from the top-ranked documents to formulate a new query for a second round retrieval, is a technique commonly used to improve retrieval performance.

We converted topics from TREC format to Indri structured queries as follows.

For automatic runs, we only used the title field of each topic. We parsed the title field, removed the stop words using a stop words list provided by Indri Search Engine, and combined the residual words by Indri operator #combine. For example, topic 804, "ban to human cloning”, after being removed stop word “to”, can be converted into the following query:

```
#combine ( ban  human  cloning )
```

For manual runs, all of the three fields of each topic were used to construct an Indri query. Our aim was to see whether manual run would perform better than automatic run. We observed the three fields of each query, selected important words based on our own knowledge, finally combined the selected words by #combine operator. We weighted each word equally. Take topic 804 for example again, we finally got a query as follows:

```
#combine(ban human cloning resolutions legislation rationale)
```

Our dependence model was only used to the title-only runs. We also removed stop words in this step. The following query is an example of queries in dependence model:

```
#weight(  0.8 #combine(ban human cloning)
          0.1 #combine(#1(ban human cloning)
                       #1(ban human)
                       #1(human cloning))
          0.1 #combine(#uw12(ban human cloning)
                       #uw8(ban human)
                       #uw8(ban cloning)
                       #uw8(human cloning) ) )
```

We adopted the default pseudo-relevance feedback algorithm in Indri Search Engine. Three of our runs made use of this technique with fbDocs=10 and fbTerms=30. The original and expanded queries were weighted equally.

Smoothing plays a very important role in language modeling technique. Indri provides several smoothing methods. We used Dirichlet smoothing for all our ad hoc runs with μ=1600 for query likelihood and μ=4500 for dependence model, using GOV2 collection to estimate parameters. For named page finding task, the smoothing parameter will be described in Section 5.

4.2 Named page finding task

In this task, we aimed to investigate the document structure and document prior. So we indexed different fields of documents, such as title, mainbody, heading, and inlink fields. We also investigated Pagerank as the document prior.

For this task, we submitted three runs. One run(TWTB06NP01) made use of all the four indexed fields, one run(TWTB06NP02) used the title field, and another one(TWTB06NP03) used the title field and Pagerank. As an example, we list three query formulations of topic NP903 in Figure 1, each of which was a formulation used for one of the three runs.

```
<top>
<num> Number: NP903
<title> reasons to reduce waste
</top>
```

(a) Topic NP903

```
#combine(#wsum( 1 reasons.(inlink)
                1 reasons.(title)
                3 reasons.(mainbody)
                1 reasons.(heading) )
         #wsum ( 1 reduce.(inlink)
                1 reduce.(title)
                3 reduce.(mainbody)
                1 reduce.(heading) )
         #wsum ( 1 waste.(inlink)
                1 waste.(title)
                3 waste.(mainbody)
                1 waste.(heading)  )  )
```

(b) query formulation used for TWTB06NP01

```
#combine( reasons.(title)  reduce.(title)   waste.(title) )
```

(c) query formulation used for TWTB06NP02

```
#weight(0.1 #prior(pagerank)
        1  #combine( reasons.(title)  reduce.(title)  waste.(title)))
```

(d) query formulation used for TWTB06NP03

Figure 1: Topic NP903 and its corresponding three Indri query formulations

5. OFFICIAL RUNS

We submitted 5 runs for the ad hoc task, two of which were manual runs and the other three were automatic runs, and submitted 3 runs for the named page finding task.

For the ad hoc task, we submitted:

TWTB06AD01: Automatic title-only run using query likelihood, dependence model, and pseudo-relevance feedback.

TWTB06AD02: Manual run using query likelihood and pseudo-relevance feedback.

TWTB06AD03: Manual run using query likelihood only.

TWTB06AD04: Automatic title-only run using query likelihood and dependence model.

TWTB06AD05: Automatic title-only run using query likelihood as our baseline run.

For the named page finding task, we submitted:

TWTB06NP01: A run using title, mainbody, heading and inlink fields. We smoothed the language model with μ=10, 40, 100, 250 for title field, heading field, inlink field and mainbody field respectively.

TWTB06NP02: A run using the title field of the documents. The smoothing parameter μ equaled 10.

TWTB06NP03: A run using the title field and pagerank. The smoothing parameter μ equaled 10.

All 8 runs submitted are summarized in table 1 and 2.

Run	P@5	P@10	MAP	R-Precision	bpref
TWTB06AD01	0.5720	0.5480	0.3737	0.3938	0.4193
TWTB06AD02	0.5280	0.5220	0.3152	0.3413	0.4089
TWTB06AD03	0.5720	0.5340	0.3033	0.3433	0.4027
TWTB06AD04	0.6040	0.5760	0.3563	0.3882	0.4103
TWTB06AD05	0.5240	0.5080	0.3067	0.3539	0.3667

Table 1: Runs submitted for TREC Terabyte ad hoc task using TREC topics 801-850 and top 10,000 documents.

Run	MRR	Top10	Top10 %	Not Found	Not Found %
TWTB06NP01	0.116	33	18.2	94	51.9
TWTB06NP02	**0.238**	62	34.3	80	44.2
TWTB06NP03	0.234	63	34.8	80	44.2

Table 2: Runs submitted for TREC Terabyte named page finding task using topics 901-1081 and top 1,000 documents.

Since the submitted automatic runs gave the top 10.000 documents for both topics 701-800 and new topics 801-850, TREC gave each run two evaluation results. One was for the new topics, and the other one was for all 150 topics. Here we only list the results for new topics 801-850.

From table 1, we can see that manual runs(TWTB06AD02 and TWTB06AD03) performed the same as the baseline run, and TWTB06AD02 , a manual run with pseudo-relevance feedback, performed the same as TWTB06AD03 which was a manual run without pseudo-relevance feedback. TWTB06AD01 and TWTB06AD04 were two best automatic runs, which provided a great increase over the baseline run.

From table 2, we can see that our three named page finding runs performed not very well. TWTB06NP03 which used title field and Pagerank prior gained no improvement compared with TWTB06NP02 which used only the title field of documents, and TWTB06NP01 which used all of the three fields of documents was the worst.

6. CONCLUSIONS

We participated in two tasks, submitted 5 runs for ad hoc task and 3 runs for named page task respectively, and gained much experience in the procedure of evaluations.

The problems we encountered were the low speed when retrieving the documents on a single PC and the transformation from the query format given by TREC to the format of Indri structured query. Since we used a single PC, the speed was very low. There were some characters which were reserved or not recognized by Indri structured query language, so we had to remove these characters from the query.

7. ACKNOWLEDGMENTS

This work described therein is supported by NSFC grant 60435020.

8. REFERENCES

[1] Chinese Web Information Retrieval Forum. http://www.cwirf.org

[2] J. Allan, J. Callan, K. Collins-Thompson, B. Croft, F. Feng,D. Fisher, J. Lafferty, L. Larkey, T. N. Truong, P. Ogilvie, L.Si, T. Strohman, H. Turtle, and C. Zhai. The Lemur toolkit for language modeling and information retrieval. http://www.cs.cmu.edu/˜lemur.

[3] J. M. Ponte , W. B. Croft. A language modeling approach to information retrieval. The 21st Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval , Melbourne, 1998.

[4] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In Proceedings of SIGIR 2005, pages 472-479, 2005.

[5] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval.Proc. of SIGIR 2002, New Orleans, U.S.A., pages 111 – 119.