Indri at TREC 2008: Million Query (1MQ) Track

Bob Armstrong, Xing Yi and James Allan
Center for Intelligent Information Retrieval, Department of Computer Science
University of Massachusetts, Amherst, MA 01003-4610, USA

Abstract

This work details experiments carried out using the Indri search engine for the *ad hoc* retrieval task in the TREC 2008 Million Query Track. We investigate comparing baseline runs on a stopped and unstopped corpus to a run utilizing a Dependence Model based on proximity features and a run using inferior smoothing through an intentionally poor choice of μ. The paper presents an evaluation of the results of these four different approaches.

1 Introduction

Last year a new track - Million Query (1MQ) Track was introduced for two purposes: (1) investigating which approach is better for system evaluation - building test collection from very many very incompletely judged topics or from traditional TREC pooling; and (2) exploring *ad hoc* retrieval on a large corpus. For the *ad hoc* retrieval task, each participant is required to submit results of running 10,000 given queries against the GOV2 corpus. Our search engine, Indri[1](Strohman et al., 2005) was utilized for this task. As evidenced by previous Terabyte Track results (Metzler et al., 2006), Indri is highly efficient and effective. As in 2007 (Yi and Allan, 2007), our goal this year was to provide a range of runs for use in the document selection process of the track. We did not carry out experiments intended to improve our results. Instead, we used methods that we had tried in earlier tracks. To ensure that some runs were of lower quality, we included a run this year where we intentionally selected a parameter value that we knew was sub-optimal. The resulting four runs range from excellent to mediocre according to the track evaluations.

This paper describes our experiments in detail.

2 Ad Hoc Task

For the *ad hoc* retrieval task this year, we submitted results of four automatic official runs.

We followed our previous successful approach of using proximity information in Terabyte Track (Metzler et al., 2006), and preprocessed the GOV2 collection in a similar setting. We built two GOV2 indexes, both with no special document or link structure indexing. One index stemmed all documents using the Porter stemmer and did no stopping. The other stemmed all documents using the Krovetz stemmer and stopped all documents. We ran baseline queries against both indexes. We also performed a Dependence Model run on the unstopped collection and did an experimental run against the stopped index where we set the Dirichlet smoothing parameter $\mu = 1$, a value known to be suboptimal.

2.1 Baseline - Simple Query Likelihood

Our baseline run this year, ind25QLnST08, is a simple title-only query likelihood run on the unstopped GOV2 collection. For example, topic 10001, ”comparability of pay analyses”, is converted into the following Indri query:

```
#combine( comparability of pay
analyses ),
```

which produces results rank-equivalent to a simple query likelihood language modeling run. We utilized Dirichlet smoothing and set $\mu = 1500$ without tuning.

[1] Available for download at: http://lemurproject.org/indri/

2.2 Dependence Model

In 2006's Terabyte Track, we found term proximity features were very useful for the *ad hoc* retrieval task on large scale, noisy web collection (Metzler et al., 2006). Therefore in this run, indri25DM08, we again used the dependence model (Metzler and Croft, 2005), which assumes query term order and proximity are very important for finding relevant documents. From three variants of dependence model (Metzler and Croft, 2005), we have used the sequential dependence version instead of the full dependence model because some topics have too many terms which results in very long Indri queries which are hard to run in limited time.

To give an idea of how the sequential dependence model translates topic terms into Indri queries, we give the following example, again for topic 10001:

```
#weight( 0.8 #combine(
comparability of pay analyses )
0.1 #combine( #1( pay analyses )
#1( of pay ) #1( comparability
of ) ) 0.1 #combine( #uw8( pay
analyses ) #uw8( of pay ) #uw8(
comparability of ))).
```

In this run, Dirichlet smoothing is used with $\mu = 1500$ for single term and $\mu = 4000$ for proximity features without tuning.

2.3 Simple Query Likelihood with stopping

This run, indriQLST08, is a simple title-only query likelihood run on the stopped GOV2 collection. The queries used are identical to the those in the baseline run.

2.4 Simple Query Likelihood with stopping and minimal smoothing

This run, indriLowMu08, is again a simple title-only query likelihood run on the stopped GOV2 collection but with the Dirichlet smoothing parameter μ set to 1 to produce the generally poor results of small documents. The queries used are identical to the those in the baseline run.

3 Results

The results from our four official runs are evaluated by two different approaches: NEU-style (AP-stat) and UMass-style (MTC). The corresponding estimated mean average precision (MAP) results are given in Table 1. The confidences of pairwise differences between four runs are calculated by the MTC evaluation, and given in Table 2.

RunID	NEU-style	UMass-style
ind25QLnST08	0.3001	0.0966
indriQLST08	0.2912	0.0950
indri25DM08	**0.3436**	**0.1013**
indriLowMu08	0.2164	0.0644

Table 1: Estimated MAPs by different evaluation styles, Bold figures show our best official run by each evaluation style.

Pairwise of RunIDs	Confidences
P(indriLowMu08<indriQLST08)	1.0000
P(indriLowMu08<ind25QLnST08)	1.0000
P(indriLowMu08<indri25DM08)	1.0000
P(indriQLST08<ind25QLnST08)	1.0000
P(indriQLST08<indri25DM08)	1.0000
P(ind25QLnST08<indri25DM08)	1.0000

Table 2: Confidences for Pairwise Performance Differences by the UMass-style evaluation

In Table 1, indri25DM08 is the best of four runs in both evaluation approaches. This result shows that proximity features are useful for the *ad hoc* retrieval task on large scale, noisy web collection, which is consistent with our previous finding in Terabyte Track (Metzler et al., 2006). Using proximity features shows a significant improvement with high confidence, as can be seen in Table 2.

The low μ run is by far the worst of the four runs, also in both evaluation approaches.

In this experiment, we found that Porter stemming the non-stopped index produced significantly better results than Krovetz stemming the stopped index. This matches our local experiments on the previous TREC Web Track ad hoc retrieval task in 2000 and 2001. The TREC topics and relevance judgements of experiments using the WT10G corpus can be found on the webpage http://trec.nist.gov/data/webmain.html. We have run TREC9 topics (451-500, title-only) against the WT10G Web corpus (stopped index, Krovetz stemmer), and tuning the Dirichlet smoothing parameter $\mu = 1000$, run TREC10 topics (501-550, title-only) against the WT10G Web corpus with the tuned

WT10G	MAP	b-pref	NDCG
TREC 9,stopped,Krovetz,$\mu = 1000$	0.1467	0.1455	0.3264
TREC 10, stopped, Krovetz,$\mu = 1000$	0.0677	0.0967	0.1990
TREC 9, unstopped, Porter,$\mu = 2000$	0.2044	0.2036	0.4210
TREC 10, unstopped, Porter,$\mu = 2000$	0.1937	0.1773	0.4475

Table 3: Comparing stopped and stemmed results

parameter, and then repeated the same experiments with the Porter stemmed, unstopped WT10G Web corpus, tuning the smoothing parameter $\mu = 2000$. The results are shown in Table 3. [above]

4 Conclusion

This year in the *ad hoc* retrieval task of Million Query Track we investigated how the Indri search engine performs with large number of queries in noisy web environments. We submitted four official runs to explore the effect of using proximity features and of using a poor Dirichlet smoothing μ for this task. Positive results were obtained by using proximity features and dependence modeling, while, as expected, poor smoothing produced poor results. Also, when doing IR on a Web corpus, consistant with previous work, our results show that it may be a better choice to use Porter stemming on unstopped data.

Acknowledgments

We thank Don Metzler and Trevor Strohman for their help in producing the queries used in this evaluation. This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

References

Xing Yi and James Allan. 2007. Indri at TREC 2007: Million Query (1MG) Track. In *Proceedings of 2005 Text REtrieval Conference (TREC 2007)*.

D. Metzler and W.B. Croft. 2005. A markov random field model for term dependencies. In *Proceedings of SIGIR 2005*, pages 472–479.

D. Metzler, T. Strohman, Y. Zhou, and W.B. Croft. 2006. Indri at TREC 2005: Terabyte track. In *Proceedings of 2005 Text REtrieval Conference (TREC 2005)*.

T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. 2005. Indri: A language model-based serach engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*.