Indri at TREC 2007: Million Query (1MQ) Track

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 the experiments carried out using the Indri search engine for the *ad hoc* retrieval task in the TREC 2007 Million Query Track. We investigate using proximity features for this task, and also explore whether using a simple spelling checker - Aspell to correct plausible spelling errors in the noisy queries could help retrieval. Results evaluated by three different approaches are presented. The strength and weakness of introducing Aspell for IR are discussed.

1 Introduction

This 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; we want to further investigate its performance with large number of queries.

In addition, because there is no quality control imposed on the 10,000 given queries, some may contain spelling errors; therefore we also utilized a simple Unix spelling checker - Aspell[2], in experiments to correct plausible spelling errors. We are interested in testing how this simple spelling check approach will work for large number of queries having typos and errors.

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. Two of them utilized a spelling checker to find plausible spelling errors and give correction suggestions.

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. First, we indexed the whole GOV2 collection with no special document or link structure indexed. Second, we stemmed all documents by using the Porter stemmer. Third, we did not stop documents at index time and did not stop query terms. Last, we used Bayesian smoothing and allowed single term and proximity features (i.e. #1, #uw8) to be smoothed differently.

2.1 Baseline - Simple Query Likelihood

Our baseline run this year, IndriQL, is a simple title-only query likelihood run. For example, topic 9101, “california department of motor and vechicles”, is converted into the following Indri query:

```
#combine( california department of
motor and vechicles),
```

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] The version is Aspell 0.50.5α. Copyright is held by Kevin Atkinson, 2000.

2.2 Simple Query Likelihood + Simple Spelling check

In this run, IndriQLSC, we utilize the Unix spelling checker - Aspell to find plausible spelling errors for each topic, then combine Aspell's correction suggestions with the title to formulate a query.

Given a topic's terms, if no errors are found, we formulated the same Indri query as in the IndriQL run; otherwise, the top three corrections suggestions by Aspell are weighted and combined with the original title terms to formulate a new Indri query by using the Indri operators "#weight" and "#syn". For example, given the topic 9101, Aspell finds a plausible spelling error "vechicles" and gives seven correction suggestions, vehicles, vehicle's, vesicles, chicle's, vehicle, vesicle's, versicle's. Then, the top three terms "vehicles, vehicle's, vesicles" are combined with the original title to formulate this Indri query:

```
#weight(0.8#combine( california
department of motor and vechicles)
0.2#syn( #1(vehicles) #1(vehicles)
#1(vesicles))),
```

where punctuation in suggested terms has been removed. The weight is fixed to be 0.2 for correction suggestions and 0.8 for the original title terms.

In experiments, plausible spelling errors have been found in 1865 of 10,000 topics. Dirichlet smoothing is used with $\mu = 1500$ without tuning.

2.3 Dependence Model

In last year'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, IndriDM, we keep using 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 one because some topics have too many terms (e.g. topic 653 has 23 terms), thus the full dependence model will obtain 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 9101:

```
#weight(0.8#combine( california
department of motor and vechicles
) 0.1#combine( #1(and vechicles)
#1(motor and) #1(of motor)
#1(department of) #1(california
department)) 0.1#combine( #uw8(and
vechicles) #uw8(motor and)
#uw8(of motor) #uw8(department
of) #uw8(california department))).
```

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

2.4 Dependence Model + Simple Spelling check

In this run, IndriDMCSC, we utilize not only the spelling checker Aspell to find plausible spelling errors in each topic title, but also sequential dependence model to use proximity information.

Given a topic title, first use Aspell to check spelling errors. If no errors are found, use the sequential dependence model to transform the title to an Indri query same as in the IndriDM run; otherwise, each error term in Indri query obtained by the sequential model is replaced by an Indri operator "#wsyn()", which weights and combines the original error term and the top three correction suggestions by Aspell. We use topic 9101 again as the example. The error term "vechicles" and the top three suggestions (vehicles, vehicle's, vesicles) are combined to form:

```
#wsyn(1.0 vechicles 0.2 vehicles
0.2 vehicles 0.2 vesicles),
```

which is then used to replace every "vechicles" in the sequential dependence model Indri query, thus resulting in the final complicated Indri query:

```
#weight(0.8#combine(california
department of motor and
#wsyn(1.0 vechicles 0.2 vehicles
0.2 vehicles 0.2 vesicles))
0.1#combine(#1(and #wsyn(1.0
vechicles 0.2 vehicles 0.2
vehicles 0.2 vesicles)) #1(motor
and) #1(of motor) #1(department
of) #1(california department))
0.1#combine(#uw8(and #wsyn(1.0
vechicles 0.2 vehicles 0.2
vehicles 0.2 vesicles))
#uw8(motor and) #uw8(of
motor) #uw8(department of)
#uw8(california department))).
```

In this run, Dirichlet smoothing is used with $\mu = 1500$ for single term and $\mu = 4000$ for proximity features without tuning. Again, plausible spelling errors are found in 1865 topics, thus IndriDMCSC and IndriDM are different in 1865 queries.

3 Results

The results from our four official runs are evaluated by three different approaches: NEU-style, UMass-style, and using topics and relevance judgments in the previous Terabyte Track[3](called TBTrack-style later). The corresponding mean average precision (MAP) results are given in Table 1. The confidences of pairwise differences between four runs are calculated by the UMass-style evaluation, and given in Table 2.

In Table 1, IndriDM is the best, or the second best of four runs, by different 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). However when evaluating using large number of topics, using proximity features are not significantly better than not using them: in Table 2, P(IndriDM<IndriQL)= 0.6104; in Table 1, both the NEU-style and the UMass-style evaluations rank IndriQL>IndriDM.

It can be observed in both Table 1 and 2 that average retrieval performances have been hurt a little when using a simple spelling checker for this task: IndriQL is better than IndriQLSC, IndriDM is better than IndriDMCSC. To show the bias of choosing topics for judging does not cause this happening, we present the number of topics that have been judged by NEU and UMass, and the number of judged topics that may contain spelling errors in Table 3, which indicates Aspell did affect performances of about 16% judged topics.

To investigate the causes of this failure, we look into several specific topics listed below, in which Aspell found plausible spelling errors:

RunID	NEU-style	UMass-style	TBTrack-style
IndriQL	**0.3086**	**0.0963**	0.2960
IndriQLSC	0.3040	0.0954	0.2939
IndriDM	0.3059	0.0962	**0.3238**
IndriDMCSC	0.2981	0.0945	0.3197

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

Pairwise of RunIDs	Confidences
P(IndriDMCSC<IndriQLSC)	0.9955
P(IndriDMCSC<IndriQL)	1.0000
P(IndriDMCSC<IndriDM)	1.0000
P(IndriQLSC<IndriQL)	1.0000
P(IndriQLSC<IndriDM)	0.9909
P(IndriDM<IndriQL)	0.6104

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

`Topic169` - *hurricain prediction season 2006*

Aspell suggestions: hurricain→hurricane, hurricanes, harridan

`Topic133` - *diltiazem xr*

Aspell suggestions: diltiazem→dualism, dulcias, dillies; xr →zr, xor, xe

`Topic863` - *symptoms of adhd*

Aspell suggestions: adhd→add, ashed, dad

The estimated average precisions by the NEU-style evaluation of these three topics are shown in Table 4. It can be seen that in Topic 169, by using spelling checker to correct *hurricain* to *hurricane*, we improve the AP drastically: IndriQLSC achieved 870% improvement, compared with IndriQL. However, as shown in Topic 133 and 863, if there are proper nouns that are very important to find relevant documents, Aspell would mistakenly attempt to correct these terms, thus decreasing the IR performance a lot, with respect to the absolute AP values. Another example is that Aspell always thinks *Los Angeles* misspelled, and suggests correct *Los* to be *laos*, *leos*, or *lois*. This happens in many noisy web queries, therefore the approach of simply applying Aspell on each query hurts the performance. Although Aspell works for some topics, it is an open issue how to avoid applying this spelling checker on some topics containing proper nouns.

[3] http://www-nlpir.nist.gov/projects/terabyte/

	# of Topics	# of Topics having errors
Overall	10,000	1865
only NEU Judged	548	80
only UMass Judged	429	73
Mix Judged	801	126

Table 3: Number of topics and topics having plausible spelling errors

	Estimated APs		
TopicID	169	133	863
RunID			
IndriQL	0.0072	**0.2672**	**0.7962**
IndriQLSC	**0.0682**	0.1253	0.4699
IndriDM	0.0012	0.0410	0.7249
IndriDMCSC	0.0539	0.0347	0.7238

Table 4: Spelling Checkers' Impact on Estimated APs of Topics 169, 133 and 863 by the NEU-style evaluation. Bold figures show our best official run for each topic.

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 simple spelling checker for this task. Positive results were obtained by using proximity features and dependence modeling, while the simple approach of using spelling checker to correct topic terms failed, at least in part because many topics contain proper nouns.

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 and 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 authors' and do not necessarily reflect those of the sponsor.

References

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.*