RMIT @ TREC 2016 Dynamic Domain Track: Exploiting Passage Representation for Retrieval and Relevance Feedback

Ameer Albahem
ameer.albahem@rmit.edu.au

Damiano Spina
damiano.spina@rmit.edu.au

Lawrence Cavedon
lawrence.cavedon@rmit.edu.au

Falk Scholer
falk.scholer@rmit.edu.au

RMIT University, Melbourne, Australia

Abstract

The TREC Dynamic Domain search task addresses search scenarios where users engage interactively with search systems to tackle domain specific information needs. In our participation, we focused on utilizing passage-based representations in document retrieval and user feedback processing. In addition, we submitted a baseline retrieval method and a manual run that considers only relevant documents in the top 1000 retrieved documents. Results show that the passage based representation is inferior to the baseline method but differences are not statistically significant in terms of the Cube Test and the Average Cube Test metrics.

1 Introduction

The TREC Dynamic Domain (DD) Track [7] assumes a search scenario where a user uses a search system and provides feedback to address domain specific and diverse information needs earlier in the interaction. In this task, a program called *JIG* acts as a simulated user to give feedback about the retrieved documents to the system. The system consumes the feedback and decides to terminate the search or provide more documents to the user. In each iteration, the system is allowed to return up to 5 documents. The final system output is then judged using the Cube Test evaluation measure [4]. In addition to the Cube Test, the track also uses the Precision at Recall [2] and Expected Reciprocal Rank [3] evaluation metrics.

We submitted a total of four runs. The first is a baseline retrieval method based on a document language model [5] as implemented in Apache Solr[1]. The second is a manual run that filters out non-relevant documents from the top 1000 documents. The remaining two runs utilize a passage representation in document ranking and query expansion. Our hypothesis is that, given that the interactive relevance feedback is given at passage-level, using a passage-based representation would make an effective use of the relevance feedback.

[1] http://apache.mirror.serversaustralia.com.au/lucene/solr/6.2.0

In the following, we first describe the submitted runs in Section 2, then we detail the experimental setup in Section 3. We report the results in Section 4 with discussion in Section 5. Section 6 concludes this paper and outlines future work.

2 Methods

For every run, we execute 10 iterations and return the top 5 unseen documents in each iteration. The following describes the methods used to generate the runs.[2]

1. **rmit-lm**: In this method, we used the language modeling approach as implemented in Apache Solr[3] using Dirichlet smoothing and default parameters. For each iteration, we return the next 5 documents in the list.

2. **rmit-lm-oracle-1000**: We use the document language model to retrieve the first 1000 documents, then we use the ground truth to remove non relevant documents from the initial list of documents. For each iteration, we return the next 5 relevant documents in the relevant document list. A document is relevant if it was found as relevant in the topic's list of judged documents. Note that only global topic relevance was considered.

3. **rmit-lm-psg-max** We split documents into half overlapped passages with a passage size of 200 words and index them in Apache Solr. We then score documents based on the maximum of their passage level relevance scores. In particular, given a document d and query q, we score d using Equation 1:

$$score(q, d) = max_{p \in d_{psg}} rel(q, p) \tag{1}$$

   where d_{psg} is a list of passages generated as described above for document d and $rel(q, p)$ is calculated using a passage language model. In initial experiments using the TREC DD 2016 collections, we tested different aggregation methods such as the sum and average of scores, but the maximum produced the best results.

4. **rmit-lm-rocchio-Rp-NRd-10** This run is inspired by recent work [1] that exploits negative and positive feedback based on different document representations to improve ad hoc retrieval. In this run, we use the baseline method to retrieve the top 5 documents in the first iteration. In the following iterations, we use the Rocchio algorithm [6] to reformulate the previous query using the feedback provided by the JIG program from the previous iteration. To represent relevant documents, we concatenate relevant passages from relevant documents into a pseudo-relevant passage (Rp), whereas we use the content of the non-relevant documents as the

[2] In the descriptions submitted to the TREC DD submission system, we reported using a combination of a unigram query and bigram phrases queries in ranking documents. However, we discovered a bug in the implementation which was causing the used search engine (Apache Solr) to rank documents based on only unigram queries. This is also applied to the other runs. In addition, the manual run's submitted file didn't include results for topic number DD16-07

[3] http://apache.mirror.serversaustralia.com.au/lucene/solr/6.2.0

non-relevant units of Rocchio (NRd). Lastly, we use the top 10 non-negative terms, measured by TFIDF, from the new query vector generated by Rocchio to build the new query. We set the Rocchio parameters to $\alpha = 1$, $\beta = 0.75$ and $\gamma = 0.25$. We experimented with various combinations of representations of relevant and non-relevant units such as the whole content and the search keyword snippets from non-relevant documents, but the aforementioned representation produced the best results. We also tried different numbers of terms to form the new query such as 10, 20, 30, 100 and 1024 terms, but using 10 terms performed the best.

3 Experimental Setup

In all runs, we used the TREC Dynamic Domain 2016 dataset. The dataset consists of two topic sets from two domains. We indexed each dataset using Apache Solr[4] and ran experiments on each index separately. In both datasets, we stripped out all HTML tags and used Boiler Pipe[5] to extract the main content. We then used Solr's English analysis to process the extracted text.

Duplicate Removal

We noticed that there are many duplicate documents in the search results. Since we want to retrieve documents as soon as possible, duplicate removal is necessary. To tackle this, we used Solr's duplication component[6] to generate document signatures, iterated through duplicated signatures, and removed the duplicate documents from the index. In addition, documents that are duplicates and found in the ground truth were retained. Table 1 describes the number of documents and duplicates in each domain.

Table 1: Statistics of the TREC DD 2016 datasets used in our runs.

Domain	Number of Documents	Number of Duplicates
Ebola	194,481	23,496
Polar	244,536	11,593

4 Results

Table 2 and Table 3 show the results of the different runs at iterations 1 and 2 respectively, whereas Figure 1 and Figure 2 show the performance of the different methods at different iterations using the Cube Test (CT) [4] and the Average Cube Test (ACT) [7] respectively. The min, max, median and avg runs are the minimum, maximum, median and mean of all runs submitted to TREC Dynamic Domain submission system respectively.

[4]http://apache.mirror.serversaustralia.com.au/lucene/solr/6.2.0
[5]https://github.com/kohlschutter/boilerpipe
[6]using TextProfileSignature implementation

From the tables and figures, several observations can be seen. First, the passage based representations do not beat the baseline method (rmit-lm) as measured by CT. In particular, ranking documents using passage retrieval is inferior to the rmit-lm, average and median runs. Second, the manual run (rmit-oracle-lm-100) is actually the best scoring run. Nevertheless, looking at its performance at different iterations, it can be seen that it is not always the best; see iteration 4 onward in Figure 1. However, it is the best in ACT (Figure 2).

The third observation is the use of query expansion as reported in Table 3. We can see a slight improvement over the baseline method (rmit-lm) in the ACT, but a degradation in CT. The fourth observation is that the performance measured in CT and ACT decreases as many iterations are run for all methods.

Table 2: Runs at iteration 1. * indicates a statistically significant improvement over the baseline method (rmit-lm) using a paired t-test with $p < 0.05$

Run ID	ACT	CT
min	0.0197	0.0291
max	0.3322	0.4176
avg	0.1473	0.2051
median	0.1516	0.2174
rmit-lm	0.1857	0.2526
rmit-lm-psg.max	0.1260	0.1758
rmit-oracle.lm.1000	0.3322*	0.4176*

Table 3: Runs at iteration 2. * indicates a statistically significant improvement over the baseline method (rmit-lm) using a paired t-test with $p < 0.05$

Run ID	ACT	CT
min	0.0274	0.0438
max	0.2899	0.2643
avg	0.1379	0.1425
median	0.1452	0.1469
rmit-lm	0.1614	0.1539
rmit-lm-psg.max	0.1178	0.1275
rmit-lm-rocchio.Rp.NRd.10	0.1644	0.1439
rmit-oracle.lm.1000	0.2899*	0.2643*

5 Discussion

In general, the results show that using passage representations leads to a fall in effectiveness with respect to the document-level baseline method. However, these differences are not significant. That means we have a high variety on their performances across different queries. It will be interesting to investigate what factors affect the performance for the individual queries.

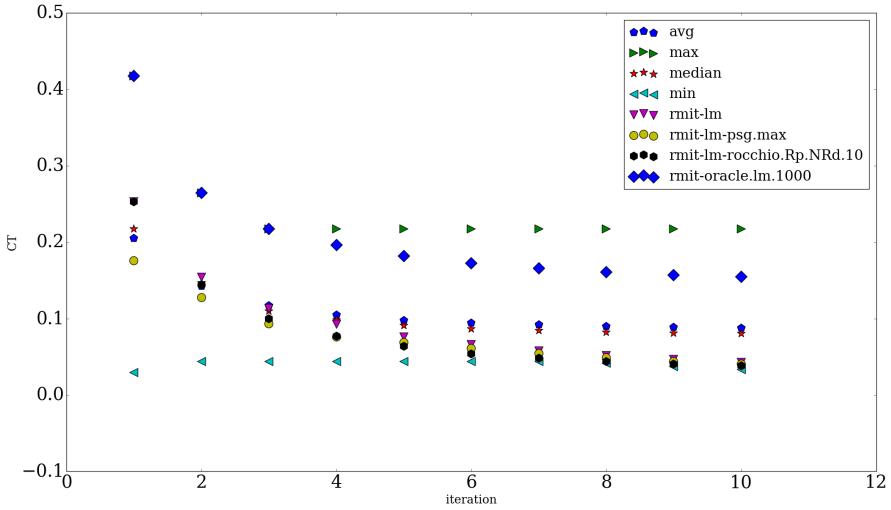

Figure 1: CT as number of iterations increases.

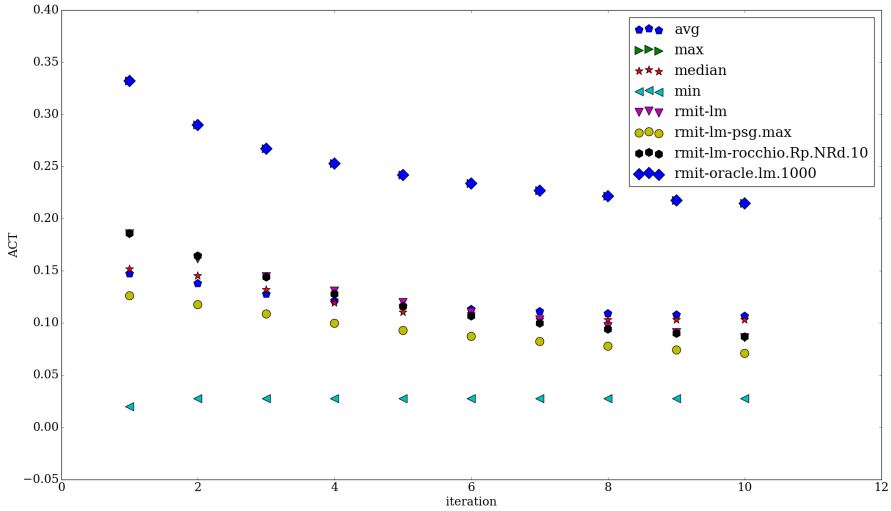

Figure 2: ACT as number of iterations increases.

The strictly decreasing performance of all runs (in particular the oracle method) across iterations indicates that it is more appropriate to compare run performances at each iteration separately than expecting to have higher scores at subsequent iterations. The ACT and CT are new measures and understanding their behavior at different iterations helps in interpreting their results; we plan to study that in future work.

The performance of our submitted runs against the mean, median and minimum of all runs submitted to the TREC Dynamic Domain might be due to the fact our duplicate removal was biased toward relevant documents. As a result, we also ran rmit-lm against an index with a blind duplicate removal (remove all duplicates without regard to their relevance), and without duplicate removal.

Table 4: Performance of the baseline (rmit-lm) with and without duplicate removal. No Removal means no duplicates were removed; Blind means duplicates were removed regardless whether they are relevant or not, and Biased means only duplicate documents not found in the ground truth file were removed.

Iteration	Duplicate Removal	ACT	CT
1	No Removal	0.1857	0.2515
	Blind	0.1807	0.2372
	Biased	0.1857	0.2526
2	No Removal	0.1632	0.1545
	Blind	0.1586	0.1492
	Biased	0.1614	0.1539

Table 4 shows that the Blind Removal is inferior to other methods that have comparable performance. We also conducted a significance test on the differences between the No Removal with the Blind and Biased removals, and found no significant difference at $p < 0.05$. The slight difference between the Blind removal and the other removals might be because it misses some of the relevant documents, which leads to differences in the scores, but this is not severe enough to cause a significant degradation.

6 Conclusion and Future Work

The TREC Dynamic Domain Track is a novel task that addresses complex search scenarios under multiple dimensions of search aspects such as search time, diversification, user feedback and relevance granularity. In our participation, in addition to a manual and a baseline runs, we submitted two runs that investigate two aspects: relevance granularity and user feedback. The former attempts to address relevance granularity by tackling the search problem as a passage retrieval, whereas the latter utilizes different combinations of document granularity (global representation and local representation using passages) to formulate new queries.

Overall, the passage retrieval approach was not as effective as document retrieval, but the difference is not significant. Utilizing passages in query expansion resulted in small improvements (but not significant) over the baseline method. In future work, we plan to investigate these differences in more detail.

Acknowledgment

This research was partially supported by Australian Research Council Project LP130100563 and Real Thing Entertainment Pty Ltd.

References

[1] E. Brondwine, A. Shtok, and O. Kurland. Utilizing focused relevance feedback. In Proceedings of the 39th International ACM SIGIR Conference

on Research and Development in Information Retrieval, SIGIR '16, pages 1061–1064, New York, NY, USA, 2016. ACM.

[2] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 33–40. ACM, 2000.

[3] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In Proceedings of the 18th ACM conference on Information and knowledge management, pages 621–630. ACM, 2009.

[4] J. Luo, C. Wing, H. Yang, and M. Hearst. The water filling model and the cube test: multi-dimensional evaluation for professional search. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pages 709–714. ACM, 2013.

[5] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 275–281. ACM, 1998.

[6] J. Rocchio. Relevance Feedback in Information Retrieval. Prentice Hall, Englewood, Cliffs, New Jersey, 1971.

[7] H. Yang, J. Frank, and I. Soboroff. Trec 2015 dynamic domain track overview.