Two selfless contributions to web search evaluation

U. Twente at TREC 2014

Djoerd Hiemstra and Robin Aly

University of Twente, The Netherlands
{d.hiemstra, r.aly}@utwente.nl

Abstract

We present our results for the Web Search track and the Federated Web Search track for the 23rd Text Retrieval Conference TREC.

1 Introduction

In this paper we describe the contributions from the University of Twente to the 23rd Text Retrieval Conference. We participate in the Web Track and the Federated Web Track. Our experiments are run by MIREX [2][1] (MapReduce Information Retrieval Experiments), a library of MapReduce programs to extract data and sequentially scan document representations. Built on Hadoop, sequential scanning becomes a viable approach. MIREX allows researchers to easily experiment with different retrieval models, because the framework is easy to extend. The paper is structured as follows: Section 2 describes our participation in the web track and Section 3 describes our participation in the Federated Web track.

2 Web Track Participation

The Web track models a general web search scenario, with ad hoc queries on the ClueWeb12 web collection, i.e., we investigate the performance of approaches that search a static set of documents using previously-unseen topics.

Table 1 shows the precision at 5, 10 and 20 results of the three official Web Track runs. The runs tagged `utbase` and `utexact` use anchor texts as document representations, which we made available for download.[2] The first run, tagged `utexact`, matches the exact query string to the anchors, and ranks the documents by the number of exact matches found. The run finds exact matches for 40 out of 50 queries. We appended the results from the second

[1] http://mirex.sourceforge.net
[2] http://www.cs.utwente.nl/∼hiemstra/2013/anchor-text-for-clueweb12.html

Run	P@5	P@10	P@20
utexact	0.456	0.412	0.375
utbase	0.440	0.422	0.371
utold	0.448	0.428	0.420

Table 1: Precision at 5, 10 and 20 (50 queries)

run, i.e. those documents that were not already found by exact matches, to the run as the final result. The second run, tagged `utbase`, uses a simple unigram language model with linear interpolation smoothing, $\lambda = 0.95$, and a document length prior. A run without smoothing (or $\lambda = 1$) retrieves the exact same top 10 documents for 45 out of 50 queries, and therefore also achieves the same precision at 5 and 10 documents. The third run, tagged `utold`, uses the same ranking as the second run on the full text of the web pages.

The experimental results show that `utexact`, exact matching of the full query string on the anchors, outperforms the other runs for precicion at 5 documents retrieved, whereas `utold`, the language model on the full text, performs best at precision at 10. Interestingly, we expected the full text run to perform much worse, because the anchor runs use a document length prior (the more anchor text, the better), that has a similar effect as an inlinks prior (also similar to PageRank).

Run	P@5	P@10	P@20
utbase	0.188	0.186	0.163
utexact	0.216	0.184	0.161
utold	0.200	0.190	0.171

Table 2: High Relevance Precision (49 queries)

Table 2 shows the ability of the systems to retrieve documents judged as *highly relevant*, *key*, or *navigational*. So, documents judged as *relevant* were not considered in this evaluation. The results show that clearly, the exact query string matching favours highly relevant documents for 5 documents retrieved.

Run	J@10	J@20	J@30
utbase	1.000	1.000	0.904
utexact	1.000	1.000	0.907
utold	0.910	0.842	0.789

Table 3: Fraction of judged documents (50 queries)

All runs were pooled to depth 20 this year, an improvement over last year's TREC when for many topics, less than 20 documents per topic were judged,

because the pool depth of 20 would have resulted in too many documents to be judged in the allotted amount of assessing time. Table 3 shows the effects of the pool depth of the fraction of judged documents for each run. The run `utold` was not part of the pool that was judged. Of the runs that contributed to the pool, at 30 documents retrieved, about 10 % of the documents is not judged. Last year, at 30 documents retrieved, judged fractions dropped between 22 % and 25 %. So, TREC seems to have done a better job judging documents this year, at least for our runs.

Qrels	**total in 2013**	**per query in 2013**	**total in 2014**	**per query in 2014**
all judged	14474	290	14432	289
highly rel. (> 1)	1106	22	1877	38
relevant (> 0)	4150	83	5665	113
irrelevant ($= 0$)	10090	202	8210	164
spam ($= -2$)	234	5	1614	32

Table 4: Number of documents judged

Table 4 shows general statistics of the TREC 2013 and TREC 2014 Web Track collections. Of the total number of documents that are judged, more than 39 % were judged relevant, so it is likely that many more relevant documents would have been found if more resources would have been available for judging.

3 FedWeb Track Participation

The Federated Web Track models a distributed search scenario where users send requests to a broker which forwards the requests to a set of search engines that are likely to produce relevant results. We participated in the resource selection task, which requires selecting resources based on resource descriptions, given a search request. The track provides sample texts and snippets form documents sampled from each search engine. Prior to resource selection, these documents have to be transformed into a resource description. Currently, resource descriptions based directly on the sampled documents in a central sample index are the most popular. However, this approach also requires substantial storage space and administrative overhead when selecting resources.

Run	**ndcg@20**	**nP@1**	**nP@5**
UTTailyG2000	0.178	0.142	0.223
Median Performance	0.182	0.202	0.250
Maximum Performance	0.460	0.534	0.601

Table 5: Official FedWeb Resource Selection results. Median and maximum performance shown for comparison.

Table 5 shows the official overall evaluation score of the run `UTTailyG2000`. Our run uses Taily [1], an approach for shard selection which showed good performance in centralized search, and adapt it for federated web search. Instead of using a centralized sample index, Taily uses vocabulary-based resource descriptions based on statistics of term related features in each shard that are used in ranking functions. Compared to this centralized setting, the full collection is only represented by a sample and and the ranking function of each individual search engine is unknown. Taily assumes a gamma distribution for scores of a query, which is inferred from the feature statistics.

4 Conclusion

We tried simple, out-of-the-box approaches to this year's Web track and Federated Web track, contributing documents to the evaluation pools, and evaluation results to the TREC report. We love TREC, and hope to be there next year with some more novel approaches to text search.

Acknowledgements

We are grateful to the European Union Project AXES (FP7-269980) and the Dutch national program COMMIT for funding our work.

References

[1] R. Aly, D. Hiemstra, and T. Demeester. Taily: shard selection using the tail of score distributions. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 673–682, 2013.

[2] D. Hiemstra and C. Hauff. Mapreduce for information retrieval evaluation: "let's quickly test this on 12 TB of data". In *Multilingual and Multimodal Information Access Evaluation*, Lecture Notes in Computer Science 6360, pages 64–69. Springer Verlag, 2010.