Tarragon Consulting at TREC 2017:

Common Core Track

Richard M. Tong
Tarragon Consulting Corporation
Berkeley, California, USA
rtong@tgncorp.com

Abstract

Tarragon Consulting Corporation (henceforth Tarragon) contributed two runs to the new Common Core track. Both were manual runs using the NIST judged topics. Both used Solr as the base search engine with the queries semi-automatically constructed from the Topic descriptions and augmented with information from Wordnet and Wikipedia.

Results are generally below the published median scores but for several topics our results are very competitive. And we did contribute a significant number of "unique relevant" documents to the judged pool.

Introduction

Our primary reason for participating in TREC this year was to respond to the track organizers desire to "… bring together the community in a common track that could lead to a diverse set of participating runs".

We have a long-standing interest in exploring whether query-augmented commercial search engines can be used effectively on TREC-style *ad hoc* retrieval tasks, so our TREC 2017 submissions are in the spirit of those previous efforts.

Our experience is that such approaches can achieve reasonable precision and often retrieve significant numbers of unique documents, although they are not usually competitive, overall, with more traditional IR methods. Given the goal of the track, this seemed like an opportunity to re-engage with TREC and contribute a potentially distinctive pair of runs.

The rest of paper is organized conventionally. We briefly describe the approach we took, highlight a few of the results, and then offer general observations.

The Basic Approach

Solr is a widely deployed open-source platform and is the search engine we use internally, so was the natural choice for this effort. It has an expressive query language that supports various forms of proximity and wildcarding, and the engine itself is very fast so enabling time-effective experimentation. We used version 6.4.1 of Solr for TREC 2017.[1]

The track data were indexed into four fields: DocId, DocTitle, DocDate, and DocBody. Text in the DocTitle and DocBody fields was lower-cased but not stemmed. Stopwords were not removed.

The topics were processed using the Stanford CoreNLP toolkit to extract named entities and part-of-speech information.[2]

We then used a three-step process to create our two submissions—we call these the FLOOR, BASE and BOOST steps.

In the FLOOR step we automatically created simple disjunctive queries from the original topics and ran these to get a preliminary pool of 10,000 documents per topic.

In the BASE step we manually created more focused queries that capture the necessary elements of the topic. The results of running these queries were merged with the corresponding FLOOR results. The BASE results were ranked ahead of the FLOOR results and the merged set trimmed to 10,000 documents. These are the *tgncorpBASE* submissions.

In the BOOST step the BASE queries were further manually augmented to include auxiliary information from the topic. The results of running these queries were merged with the *tgncorpBASE* submissions. The BOOST results were ranked ahead of the *tgncorpBASE* results and the merged

[1] https://lucene.apache.org/solr/

[2] https://stanfordnlp.github.io/CoreNLP/

set trimmed to 10,000 documents. These are the *tgncorpBOOST* submissions.

Results and Analysis

The following table shows the overall results. The scores for the two runs are the averages, and these are contrasted with the average medians for all the manual submissions on the NIST judged topics.

Run	AP	NDCG	P@10
tgncorpBASE	0.2552	0.5102	0.5260
tgncorpBOOST	0.2757	0.5354	0.5940
Median	0.3786	0.6380	0.6720

The BOOST vs. BASE strategy was generally successful, producing decent improvements on all three metrics. These are still well below the medians, however.

Nevertheless, there were a few Topics on which the approach performed very well. For example, Topic 416 (Three Gorges Project). The BASE Solr query is:

```
{!complexphrase inOrder=false}
( DocBody:("Three Gorges") ||
  DocTitle:("Three Gorges")^2)
```

This is a disjunction of the phrase "Three Gorges" in the body of the document, and the same phrase in the title, but with a title getting more weight.

The BOOST query then adds a conjunctive clause that looks for project related words in the document body. Thus:

```
{!complexphrase inOrder=false}
( DocBody:("Three Gorges") ||
  DocTitle:("Three Gorges")^2)
&&
DocBody:(complet* cost* output electric*)
```

The following table shows our performance on Topic 416 relative to the scores for all the manual submissions on this topic.

Run	AP	NDCG	P@10
tgncorpBASE	0.6706	0.8147	0.7000
tgncorpBOOST	0.7817	0.8966	0.9000
Minimum	0.5003	0.7363	0.4000
Median	0.6633	0.8550	1.0000
Maximum	0.7817	0.9118	1.0000

Given our approach, certain topics are not easily converted into augmented queries. Topics that require a determination of counts of things (e.g., Topic 626), or the identification of named entities (e.g., Topic 389), or are very broad (e.g., Topic 344), or underspecified (e.g., Topic 442), are inherently hard to express.

Our *a priori* assessment was that 20 of the 50 topics were "hard". Partitioning the results by hard vs. non-hard produces the average scores shown in the following table for the *tgncorpBOOST* submission. These are compared with corresponding average median scores for all manual submissions similarly partitioned.

Query Type	AP	NDCG	P@10
hard	0.1675	0.4149	0.4800
non-hard	0.3479	0.6157	0.6700
Median hard	0.2978	0.5798	0.6316
Median non-hard	0.4347	0.6794	0.7000

It is not an especially pronounced effect, but non-hard queries did perform better than the hard queries—both in absolute terms and with respect to their corresponding average medians.

Since our main goal was to provide potentially distinct runs, more interesting for us was to look at the "unique relevant" documents we returned. We found at least one unique document in 34 of the 50 topics, and found 203 overall.

The following table shows our top ten topics based on the proportion of the total number of unique documents found by at least one of our submissions.

Topic ID	#unique	#found	proportion
620	1	1	1.0000
416	4	3	0.7500
427	12	7	0.5833
414	2	1	0.5000
433	6	2	0.3333
336	3	1	0.3333
363	156	43	0.2756
367	12	3	0.2500
394	24	6	0.2500
347	27	5	0.1825

Comments

Overall results are consistent with our expectations. Our focus on structured queries inevitably emphasizes precision over recall, but at the same time can find documents that more conventional techniques do not.

Given our goals, we regard this as a successful participation in TREC 2017. And it was, we hope, a positive contribution to the community effort.