UMD_CLIP: Using Relevance Feedback to Find Diverse Documents for TREC Dynamic Domain 2017

Kristine Rogers and Douglas W. Oard
College of Information Studies and UMIACS
University of Maryland, College Park
krogers@umd.edu, oard@umd.edu

Abstract

The University of Maryland's participation in TREC's 2017 Dynamic Domain track focused on two types of experiments: adding new terms from passages judged as being relevant, and exclusion of terms from documents that the track's jig feedback system indicated were not relevant to the topic. The best results for iterative multi-step retrieval were obtained by restricting retrieval to documents that contained all topic terms, and then ranking those documents using terms extracted from known relevant passages.

Introduction

The participation of the Computational Linguistics and Information Processing Lab at the University of Maryland (UMD_CLIP) in TREC 2017 Dynamic Domain track (TREC-DD) focused on characterizing user intent from jig feedback, and on improving over a simple no-feedback baseline that performed a single query using TREC-DD topic terms and then sequentially returned the documents in rank order, going deeper in that list with every iteration. Our experimental conditions can be categorized in two ways: adding relevant terms, and excluding irrelevant terms. We obtained our best results from an approach that required the inclusion of the original topic terms and then ranked that subset based on the presence of terms from relevant passages, as provided by the TREC-DD jig—the system through which relevance feedback was provided.

General Approach

We implemented all of our methods using the Indri information retrieval system from the University of Massachusetts using Indri's default ranking function with stemming disabled. All of our results are for the automatic condition, with no human intervention. Each TREC-DD run[1] involves repeatedly submitting iterations of 5 document results at a time, until a total of 25 documents (5 iterations of 5 results) have been submitted. The TREC-DD track coordinators allowed up to 10 iterations per run; however, we stopped after 5 iterations as a static estimate of the state where the user's information need has been met.

[1] Here we use "run" in the usual TREC sense to mean the complete set of results on which an aggregate score is computed. Each run in our submissions contained five iterations, containing no more than five documents per iteration.

In every run, our first iteration (i.e. set of 5 document results) was obtained by using the TREC topic terms as the query, performing ranked retrieval, and submitting the top 5 results to the jig—the system that provided simulated user feedback. We then formulated the query for each subsequent set of 5 submitted document results separately, based solely on the topic terms and the 5 results reported as relevant or as not relevant by the jig for the immediate prior query. Stopwords were consistently removed from the TREC-DD topic and from documents in the result sets before queries were formed. To avoid submission of duplicates, we maintained a running list of submitted documents for each run, and any documents on that list were removed from the result set before selecting the 5 document results to submit to the jig.

Our principal focus is on Normalized Cube Test (nCT) results, the normalized version of the Cube Test (an evaluation measure used in all three TREC-DD tracks). For completeness, we also report Cube Test (CT), Average Cube Test (ACT), nCT, Session Discounted Cumulative Gain (sDCG), normalized sDCG (nsDCG), Expected Utility (EU), and normalized EU (nEU) results.

Official Runs

Official runs are those runs that are submitted on time and scored by NIST. We submitted three official runs to TREC-DD 2017:

clip_baseline: Our baseline run performed a single search using the topic terms as the query and then iteratively returned the first 25 documents, five at a time, ignoring any relevance feedback from the jig.

clip_addwords: Our addwords run used positive feedback to perform query expansion. As always, the first query was formed only from topic terms. Subsequent queries were formed from topic terms plus terms from documents reported by the jig to be relevant. Each term (from the topic or from the positive feedback expansion set) was given equal weight.

clip_filter: Our filter run used positive feedback to perform reranking. Indri's filter operator performs a Boolean conjunction (i.e., an AND operator) over the "required" term set and then reranks the remaining documents using an "optional" term set. We used the topic terms as the required term set and the terms found in the relevant document as the optional term set.

The expansion terms in our addwords run and the optional terms in our filter run were identical. Our unconstrained approach to positive feedback (removing only stopwords) resulted in term sets ranging from small numbers—less than ten—to several hundred new terms.

For the baseline run we searched the full New York Times (NYT) collection (1987-2007). Because of a configuration error, the other two official runs searched only the first half of the collection (1987-1997). We therefore focus on our baseline run results here, and we report unofficial results below for our addwords and filter runs with the entire collection.

Table 1: Baseline Results

CT	ACT	nCT	sDCG	nsDCG	EU	nEU
0.1228	0.2136	0.6215	25.9919	0.4492	30.3262	0.4487

Figure 3 is a chart drawn from the TREC-DD results at the time of the conference. The *clip_baseline* run, the only valid official run we submitted, is shown in orange (it is the best of our 3 official runs). We note with interest that, although no new results were submitted after the fifth iteration (the Cube Test score remains static), the nCT score appears to increase; the measure rewards result sets that provide relevant documents in a smaller number of iterations.

Unofficial Runs

After the submission deadline we repeated our addwords and filter runs on the entire collection, and we report those results here as locally scored unofficial runs. We also ran three additional contrastive conditions as unofficial runs. For this larger set of runs, we found it convenient to adopt a richer and more compact nomenclature: T=topic terms, N=new (positive feedback) terms, S=spam (negative feedback) terms, f=filter, w=weighted, r=reject.

Positive Feedback

T+N: This is the same as our official addwords run, but searching the entire collection.

fT+N: This is the same as our official filter run, but searching the entire collection.

wT+N: This is a variant of T+N in which we give five times as much weight to a topic term as to an expansion term.

Negative Feedback

We tried some negative feedback approaches that were motivated generally by an approach described in the TREC 2015 Dynamic Domain track overview (Yang & Soboroff, 2015) that was attributed there to Beijing University of Posts and Telecommunications (BUPT). We refer to the non-stopword terms in documents reported by the jig not to be relevant as "spam" terms. We tried two negative feedback variants:

rS+T: Indri includes a variant of the filter operator can accept list of "reject" terms. If any reject term is present in a document, that document will be removed from the result set. For our rS+T run we specified the spam terms as reject; the remaining documents were then reranked based on the topic terms.

T!S: This run (read "T not S") used Indri's fuzzy negation operator to downweight documents that contained spam terms.

Unofficial Results

As shown in Table 2, our fT+N run did much better than all of the others by the nCT measure, with statistically significant improvements over the baseline (two-tailed paired *t*-test, $p<0.05$).

The rS+T approach also yielded a statistically significant improvement (two-tailed paired *t*-test, $p<0.05$), though it had a smaller improvement in nCT scores than observed with fT+N. Here the baseline result is denoted with the letter T.

Table 2: Results from baseline, positive, and negative feedback retrieval approaches for 5 iterations.

	CT	ACT	nCT	sDCG	nsDCG	EU	nEU
Baseline							
T	0.1228	0.2136	0.6215	25.9919	0.4492	30.3262	0.4487
Positive Feedback							
T+N	0.1278	0.2195	0.6468	**30.7061**	**0.5404**	**37.6496**	**0.5064**
fT+N	**0.2667**	**0.2872**	**1.3506**	22.2979	0.3731	24.1459	0.4079
wT+N	0.1236	0.2167	0.6257	27.2080	0.4690	31.7507	0.4576
Negative Feedback							
rS+T	0.1410	0.2265	0.7133	23.2507	0.3995	26.1024	0.4223
T!S	0.1172	0.2115	0.5927	23.4830	0.4102	26.8278	0.4312

Figures 1 and 2 present comparisons of our positive and negative feedback approaches, respectively. Both graphs are sorted in descending order based on the original baseline nCT scores. Note that in Figure 3 the line for the baseline values is obscured by the lines for T+N and wT+N, which received similar nCT scores.

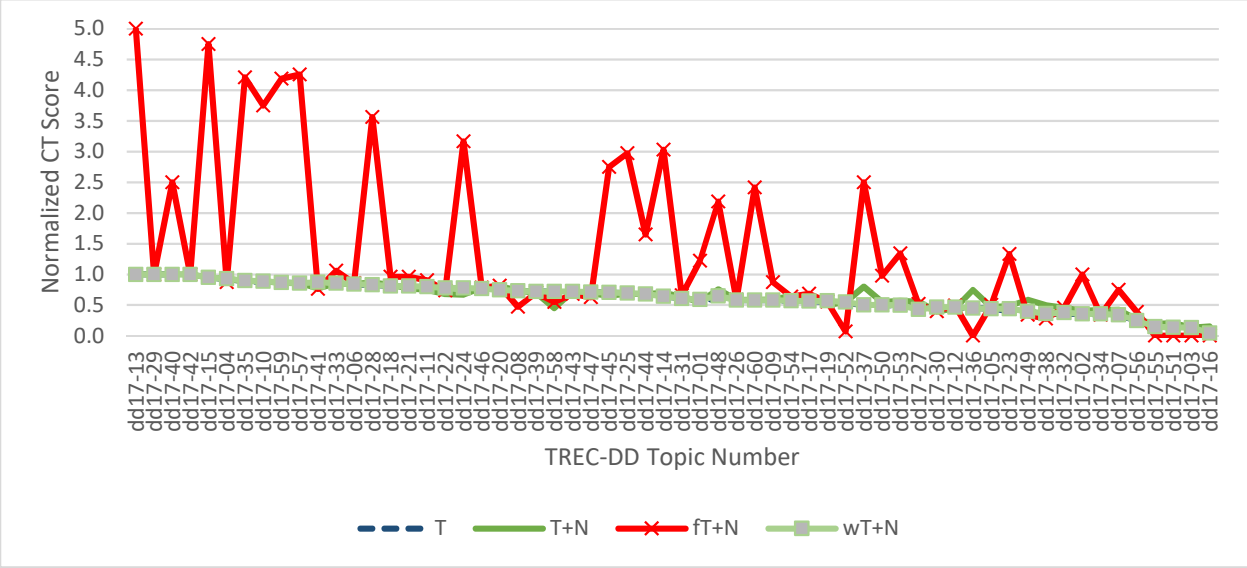

Figure 1. Comparing positive feedback approaches, sorted in descending order by baseline nCT score.

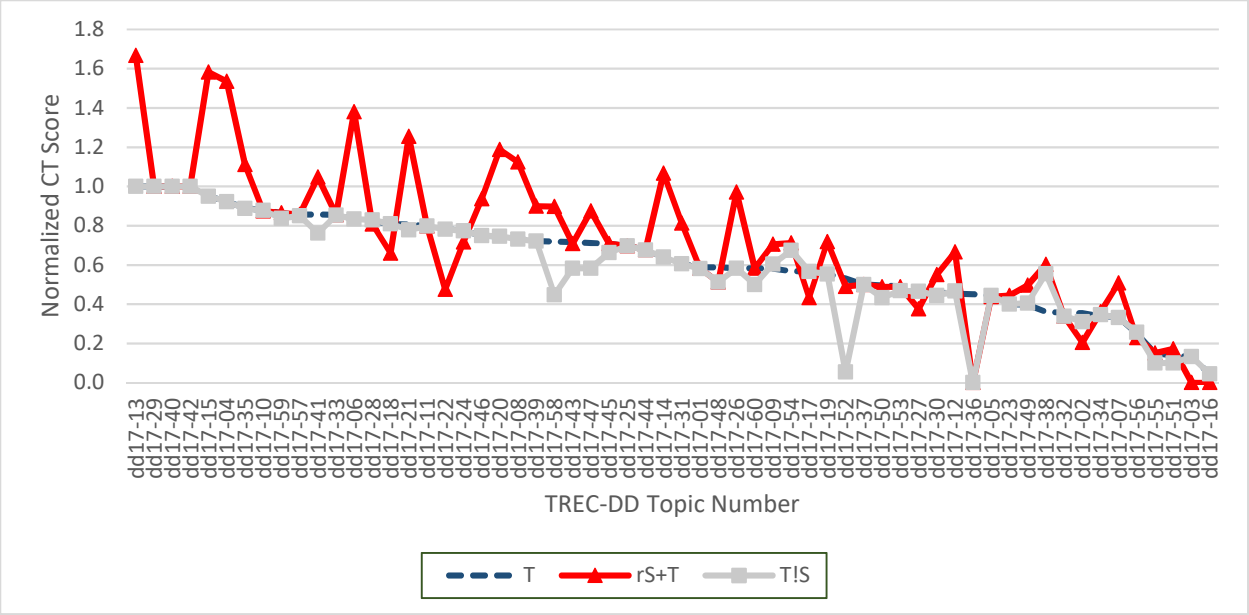

Figure 2. Comparing negative feedback approaches, sorted in descending order by baseline nCT score.

Our analysis revealed that approaches that made use of weighting produced CT that were not statistically significantly different from the baseline scores. This included assigning high weights for new terms from the jig, low weights for spam terms, and combining the two options.

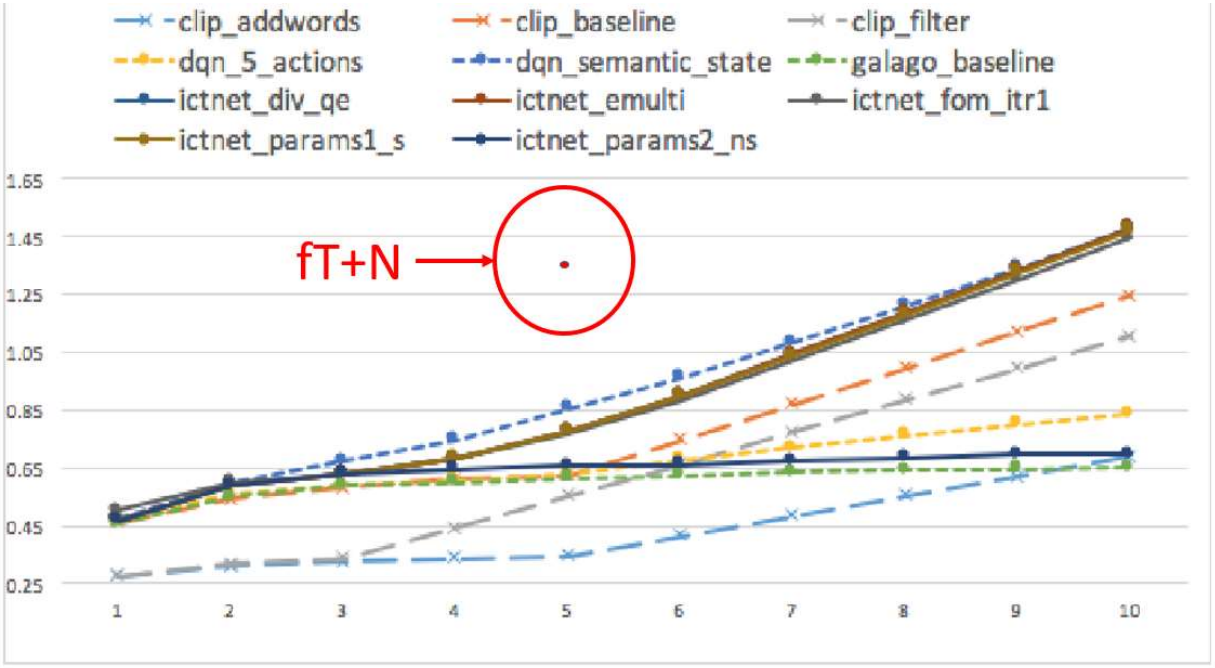

Figure 8. Normalized CT scores in the first ten iterations

Figure 3. TREC-DD 2017 official runs, with an added point showing our highest scoring unofficial run (fT+N) after 5 iterations (best viewed in color).

In Figure 3 we show where our highest scoring run—fT+N—would have fallen relative to the TREC-DD official submissions.

Future Work

We should note that aside from removing stopwords, we did not perform any cleanup of added words or spam words. Looking back at the generated query sets, there are cases where identical or variant spellings of terms were contained in the original keywords, added keywords, or spam keywords. We thus should also consider variants that make selective use of stemming. We also should ensure a careful separation between topic and feedback terms, since with our current approach, a topic term could also appear in the positive feedback.

Future extensions of our approach could also include combining the negative and positive approaches—resulting in a case where we add in relevant terms learned from the jig, while also removing documents with "spam" terms.

Other approaches for accomplishing this task could include clustering the document results for a given subtopic, then submitting the highest scoring documents from five separate clusters for each run. This could further improve the diversity of the document result sets.

Some of our runs were quite slow. For our positive feedback approaches, it took Indri 10-60 minutes to run the full set of queries. For negative feedback, Indri took 1-4 hours for each set of queries. Clearly, once we settle on an effective set of techniques, we would then want to work on efficient implementation.

Conclusion

Our participation in the 2017 TREC-DD track included experiments with Indri operators for adding and removing search terms as a method for producing a more diverse document result set. A statistically significant improvement was obtained over a simple ranked retrieval baseline using Indri's filter operator with positive feedback. Smaller but potentially promising positive effects – also statistically significant – were seen from the use of a variant of the filter operator with negative feedback.

References

Yang, G. H., & Soboroff, I. (2017). TREC 2016 Dynamic Domain Track Overview. In TREC.