WaterlooClarke at the TREC 2019 Conversational Assistant Track

Charles L. A. Clarke

School of Computer Science, University of Waterloo, Canada
claclark@gmail.com

1 Introduction

For TREC 2019, I, the WaterlooClarke group, submitted four runs to the Conversational Assistant Track (CAsT), which in combination represent three experiments:

- `clacBase` vs. `clacBaseRerank`
- `clacBase` vs. `clacMagic`
- `clacMagic` vs `clacMagicRerank`

This report details the generation of each of these runs and the outcome of these experiments.

My overall approach can be explained as three steps: 1) query construction, 2) passage retrieval and ranking, and 3) passage re-ranking. The third step is optional and applies only to the `clac*Rerank` runs. Like everyone else participating this first year, my ability to explore alternatives for these steps was hampered by the lack of training data specific to this track, and so ultimately I adopted relatively simple methods for all steps.

During an exploratory phrase, while these methods were being selected and developed, I did a fair amount of seat-of-the-pants judging and side-by-side comparison of results on selected training topics. I never want to read about physician assistants, Plessy v. Ferguson, or water molecules ever again. On the other hand, I have lots of ideas for improving these methods, and I'm looking for to the continuation of the track in TREC 2020.

2 Details of Method

2.1 Query construction

The starting point for query construction varied depending on the run:

`clacBase*` — `evaluation_topics_v1.0.txt` and `test_topics.query`
`clacMagic*` — `evaluation_topics_annotated_resolved_v1.0.tsv`

My understanding from the track website[1] is that `test_topics.query` was generated from the questions in `evaluation_topics_v1.0.txt` without direct human input. ("AllenNLP coreference resolution to perform rewriting and stopwords are removed using the Indri stopword list.") If my understanding is correct, the `clacBase*` runs may be viewed as *automatic runs* in the traditional

[1] `treccast.ai`

```
stopwords = {
    "a", "about", "an", "and", "any", "are", "as", "at", "be", "being", "by",
    "can", "defines", "describe", "description", "did", "do", "does", "for",
    "from", "give", "had", "has", "have", "his", "how", "i", "if", "in", "is",
    "isn", "it", "its", "like", "many", "may", "me", "much", "my", "of", "on",
    "once", "one", "ones", "or", "s", "should", "so", "some", "such", "t",
    "tell", "than", "that", "the", "their", "them", "there", "these", "they",
    "this", "to", "use", "using", "was", "we", "well", "were", "what", "when",
    "where", "which", "who", "why", "will", "with", "you", "your",
}
```

Figure 1: Stopword list developed from training questions

TREC sense. On the other hand, `evaluation_topics_annotated_resolved_v1.0.tsv` was created through "manual resolution of coreference as well as conversational ambiguity for topics," and so the `clacMagic*` runs should be viewed as *manual runs* in the traditional TREC sense.

Working with the equivalent training questions I explored various methods to improve on the coreference resolution, mostly using the Stanford NLP toolkit[2]. Nothing I did appeared to improve noticeably on the track queries based on AllenNLP coreference resolution. Since other participants would also be using these track queries, I decided to stick with those.

However, my seat-of-the-pants evaluation methodology suggested that the Indri stopword list used to generate the track queries was overly aggressive, at least in the context of this track. As a result, I sat down with the training questions and hacked out my own track-specific stopword list based on nothing more than a vague and entirely personal sense of word importance. For the sake of completeness, I include this stopword list as Figure 1. To apply this stopword list for the `clacBase*` runs, I first joined the original questions in `evaluation_topics_v1.0.txt` with the queries in `test_topics.query` and filtered the result against the list. For the `clacMagic*` runs, I simply filtered `evaluation_topics_annotated_resolved_v1.0.tsv` against the list.

Since tracking conversational context is essential, I also spent some time exploring various methods. Based on my seat-of-the-pants evaluation methodology nothing I tried appeared to work as well as just gluing the initial query in each conversation onto all of the following queries. This crude method was used for all runs. If more runs had been permitted, I would have submitted runs with and without this crude hack, since my guess is that it had substantial positive impact.

Again based on my seat-of-the-pants evaluation methodology, stemming appeared to do far more harm than good, so queries were executed without stemming.

2.2 Passage retrieval and ranking

Passage retrieval and ranking used standard BM25 ranking with pseudo-relevance feedback as described in Chapter 8 of Büttcher et al. [1]. BM25 retrieval used parameters of $b = 0.25$ and $k = 0.75$ for both initial retrieval and post-feedback retrieval. These parameters are based on previous tuning on ancient TREC collections. Since BM25 known to be highly sensitive to parameter tuning, these parameters are unlikely to be optimal for this collection and these queries. Pseudo-relevance

[2] `stanfordnlp.github.io/CoreNLP/coref.html`

run id	type	map@5	ndcg@5	±Δ (p-value)	
`clacBase`	automatic	0.0519	0.3504		
`clacBaseRerank`	automatic	0.0499	0.3491	−0.0013 (0.926)	vs. `clacBase`
`clacMagic`	manual	0.0629	0.4053	+0.0549 (0.010)	vs. `clacBase`
`clacMagicRerank`	manual	0.0613	0.4059	+0.0006 (0.970)	vs. `clacMagic`

Figure 2: Summary of results

feedback expansion was based on an initial retrieval of the top-32 documents, adding up to 16 terms to the final query using a mixing parameter of ($\gamma = 1/3$).

During the exploratory phrase, I experimented with other retrieval methods, particularly with passage-oriented methods developed in the context of earlier TREC questions answering tasks. Perhaps because the collection for this task has already been split into passages, these methods provided little or no obvious benefit over BM25. In addition, I made some preliminary efforts to identify and take advantage of named entities in the queries, but ran out of time.

2.3 Passage re-ranking

Passage re-ranking was used only for the `clac*Rerank` runs. Passage re-ranking was done essentially blind, since the training topics were used to generate training data for re-ranking. Apart from a couple of test questions I created myself, there was no real way to even guess if the re-ranking would do anything one way or the other.

Re-ranking was treated as a classification task with class probabilities used for re-ranking. To generate training data, I used three examples for each training question. The top passage retrieved using the query construction and passage ranking method above was treated as a positive example. The 500th passage retrieved using the query construction and passage ranking method above was treated as a negative example. A random passage retrieved by a query from a different topic was treated as a second negative example.

Each example used only the first 50 tokens of the retrieved passage. These passages were paired with the original question text, rather than the stopped queries. The text of the first turn was prefixed to the front of each question text after the first turn to maintain conversational context in a crude way. These examples were used to fine-tune BERT-Base[3] using standard parameters and class probabilities were used for re-ranking the top-32 passages from the test runs.

3 Results

Figure 2 provides basic results. P-values are based on a two-sided, paired t-test. As would be expected, the use of the manually resolved topics provided a significant and substantial improvement. Passage re-ranking provided no obvious benefit, perhaps due to the inadequacy of the training data.

[3] `github.com/google-research/bert`

4 Conclusion

The effort this year provides a reasonable starting point for next year. Improved methods for tracking and maintaining conversational context, and actually making re-ranking work, have the highest priority. Incorporating support for named entities, and better retrieval methods will be considered. Looking forward to TREC 2020 CAsT.

References

[1] Charles L. A. Clark, Stefan Büttcher, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines.* MIT Press, 2010.