MPII at TREC CAsT 2019:
Incoporating Query Context into a BERT Re-ranker

Samarth Mehrotra
Technical University of Munich
samarth.mehrotra@tum.de

Andrew Yates
Max Planck Institute for Informatics
ayates@mpi-inf.mpg.de

Abstract

MPII participated in the Conversational Assistance Track (CAsT) at TREC 2019. Our approach consists of an initial stage ranker followed by a BERT-based [3] neural document re-ranking model. BM25 with query expansion based on external knowledge (i.e., Wikipedia and ConceptNet) serves as the first stage ranking method, while the neural model uses BERT embeddings and a kernel-based ranking module (KNRM) to predict a document-query relevance score. We repurpose and modify subtopics from the TREC Web Track's diversity task to train the neural module. We find that the neural re-ranking module substantially improves upon the initial ranking approach.

1 Introduction

The Conversational Assistance Track (CAsT) is a new track introduced by TREC 2019, which focuses on information seeking in a conversational setting. MPII participated in CAsT with an approach based on paragraph retrieval: a first stage ranker with query expansion identified a set of candidate documents that were then re-ranked with a BERT-based re-ranking model. [3–5] Following [4], we used a BERT-KNRM variant [9] combining BERT's embeddings with KNRM's kernel pooling technique. The current utterance and n previous utterances serve as the query, which we concatenate before processing with an attention module. In this short paper, we present our methodology and preliminary results.

2 Methodology

In this section, we describe in detail the CAsT task and our approach. Given a sequence of questions, $q_1, q_2, ...q_n$, the objective of the CAsT task is to return a relevant list of documents for each step of the conversation. At each step i, the previously seen questions (i.e., q_1 to q_{i-1}) can be considered part of the conversation history. Documents were retrieved from a collection of three corpora: MS Marco, TREC CAR, and the Washington Post corpus from the TREC News Track. Given that this is a conversational search task, nDCG@3 is used as the main evaluation metric.

Our approach consisted of a two-stage retrieval method, as is common when employing a neural ranking method. The first stage consisted of a recall-focused approach (i.e., BM25 with query expansion). The second stage consisted of a BERT-based neural re-ranking module that re-ranks the documents retrieved by the initial ranker. Both stages are described in the following sections. Given that this was the first time the CAsT track appeared at TREC, only a small amount of human judgments were available.[1] Rather than training our neural model with a mixture of these judgments and judgments from other tasks, we rewrote existing TREC queries and used the associated judgments provided by TREC.

2.1 Query Expansion and Initial Retrieval

Given the challenges posed by conversational queries, we opted for a recall-focused first stage ranking approach using query expansion via Wikipedia and ConceptNet [6]. In pilot experiments we found that performance on TREC Sessions data and the TREC Web Track's diversity task data (with queries rewritten to be conversational) improved over BM25 with this approach.

To retrieve a set of documents for query q_i, we begin by concatenating all of the previous queries in the conversation history and the current query, i.e. from q_1 to q_i, to form a new query called q_c. Inspired by [8], we then expand q_c using a query expansion module that uses external knowledge. We use a combination of ConceptNet and Wikipedia to expand the queries with relevant terms which can potentially improve the quality of the initial retrieval. As illustrated in Figure 1, Wikipedia was used for queries that contained entities (according to Spotlight), while ConceptNet was used for queries that did not. We use DBPedia Spotlight [1] to perform entity linking on the concatenated query q_c. Related Wikipedia pages were then identified using Wikipedia's Search API. Relevant terms from these pages were then scored using a TF-IDF heuristic similar to that used in [8].

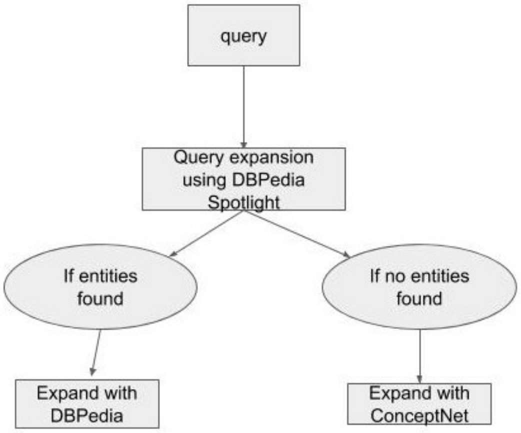

Figure 1: Overview of resources used for query expansion.

For each entity returned by Spotlight, we expand the query using terms from Wikipedia as follows. First, we query Wikipedia's search API and retrieve a list of snippets from relevant pages. Let $o_1, ..., o_n$ be the list of snippets returned. For each unique term t_i in these snippets, we compute a score for the term as follows: $score_{t_i} = \sum_{o_1}^{o_n} (tf(o_j, t_i)/|o_j|) * r(o_j) * \log(1/df(t_i))$; where $r(o_j)$ is the reciprocal rank of o_j in the results, df is the document frequency of a term, $|o_j|$ is the length of the summary and $tf(o_j, t_i)$ is the term frequency of the term in the o_j^{th} summary. Up to 10 of the highest scored terms are added to the query. If no entities are identified in the query, it is instead expanded using ConceptNet. To do so, n-grams comprised of nouns and adjectives from the query are used to query for relevant nodes. Terms from the connecting nodes are then added to the query.

[1] CAsT organizers provided thirty training topics; incomplete manual judgments were provided by five of these.

The concatenated query q_c is expanded to form the expanded query q_e, which we then use to identify candidate documents with BM25. This set of potentially relevant documents are then re-ranked as described in the next section. Note that this is a recall-oriented first stage retrieval and our second stage re-ranking method does not consider BM25's scores.

2.2 Document Re-Ranking with BERT and KNRM

Given a potentially relevant document D, our goal is to predict a relevance score between query q_i and D that takes the conversation context (i.e., queries $q_1, ...q_{i-1}$) into account. We use an approach based on the BERT-KNRM neural re-ranker [4], but use an additional attention module to re-weight embeddings from the document and query sequence before consuming them with KNRM. [9]

We incorporate past queries by concatenating the previous queries with the current query. However, at greater depths of the conversation, this can lead to long concatenated queries. Hence, we define a sliding window heuristic to not consider all the previous queries in the conversation history. We first resolve any coreference mentions in the sequence of queries. We then consider only the previous five queries and concatenate them with the i^{th} query. Important terms from distant queries (i.e., more than 5 turns back) may be incorporated due to coreference resolution, while coreference resolution failures are mitigated due to the inclusion of recent queries.

Consider the concatenated query (after coreference resolution and the sliding window heuristic) q_c and the document D. Let D consist of a sequence of terms $p_1, p_2, ...p_m$ and q_c consist of a sequence of terms $q_1, q_2, ...q_n$. We use a pre-trained BERT model to extract contextual word embeddings for each of the words in q_c and D.

Let the extracted embeddings be $e_{p1}, e_{p2}, ...e_{pm}$ and $e_{q1}, e_{q2}, ...e_{qn}$. These extracted contextual embeddings are now passed through an attention module that is used to re-weight parts of the query as well as the document. This attention module used is identical to the attention module described in the paper by [7]. As in [7], the module uses scaled dot-product attention: $Att(Q, K) = [softmax(Q[i] * K^T/\sqrt{d}]_{i=0}^{n_Q-1}$ and $V_{att} = Att(Q, K) * V$.

Let the term embeddings of query and document tokens extracted using a BERT model be:

$$doc_{embeddings} = e_{p1}, e_{p2}, ...e_{pm} \tag{1}$$

$$query_{embeddings} = e_{q1}, e_{q2}, ...e_{qn} \tag{2}$$

After using the attention module, these become:

$$query^{new}_{embeddings} = Attention(query_{embeddings}, doc_{embeddings}, doc_{embeddings}) \tag{3}$$

$$doc^{new}_{embeddings} = Attention(doc_{embeddings}, query_{embeddings}, query_{embeddings}) \tag{4}$$

The new embeddings, $query^{new}_{embeddings}$ and $doc^{new}_{embeddings}$ are passed as input to the embedding layer of the KNRM model. KNRM uses multiple kernels to consider different levels of embedding similarity in order to predict a relevance score; see [9] for further details. The only change which was made to the KNRM model was the loss function: we used binary cross entropy (as in [5]), because the training data which we artificially created was binary (relevant/non-relevant). This re-ranking approach is then used to re-rank the candidate documents returned by the first stage retrieval method.

3 Experimental Setup

As this was the first year of the track, the amount of labeled training data available was limited. However, we required training data for a number of reasons: to fine-tune the BERT embeddings, to learn the weights of the new attention layer which was added, and to learn the weights of the KNRM module.

In order to generate training data, we made use of previous collections of the TREC Web Track (diversification subtask) data.[2] The task was similar to the TREC CAsT track in the sense that it involved multiple information seeking queries which were based on a broad topic. Unlike CAsT, the queries were not very conversational and did not consist of co-references between queries. To resolve this problem and create a more appropriate training set, we manually identified the key entities being mentioned in each sequence of queries and randomly replaced certain occurrences of these entities with appropriate pronouns. This gave us a new training set which was more similar to TREC CaST.[3] For example, the following queries were part of the Web Track data:

- Find the homepage for the Kansas City Southern railroad
- I'm looking for a job with the Kansas City Southern railroad.

After replacing entities with co-references, the queries might become:

- Find the homepage for the Kansas City Southern railroad
- I'm looking for a job with them.

We used Anserini [10] to index the three collections. In the initial stage retrieval step, we retrieved 500 documents from TREC CAR, 500 from MSMarco and 200 from WaPo. In the next step, we used spaCy[4] as the tool to perform co-reference resolution. To fine-tune the BERT embeddings, we first used the fine-tuned and pre-trained embeddings which have been made available by [5]. We then fine-tuned these embeddings using data from the Sessions Track. Finally, we fine-tuned the embeddings while training the KNRM module end-to-end with the Web Track Diversity data which we converted to conversational queries.

We submitted two runs to CAsT. The first run, `mpi_base`, was a simple run which comprised only of the stage one results, i.e. query expansion and retrieval using BM25. The second run, `mpi_bert`, was comprised of the entire pipeline: initial retrieval followed by re-ranking using the neural model.

4 Results and Conclusion

The results of our approaches are shown in the top half of Table 1. The BERT-KNRM-based re-ranking model substantially improved MRR and nDCG@3 over our recall-focused first stage retrieval. The first stage retrieval method did not perform well in terms of MAP when compared to other participants, though it did outperform the BERT-based model here (due to the limited number of documents re-ranked).

In the second half of Table 1, we show results from a similar BERT-based method: `ug_cedr_rerank`. While this approach used a similar CEDR-KNRM architecture, it performed substantially better across metrics. There are at least three possible contributing factors: *(1)* prior work has indicated

[2] https://trec.nist.gov/data/webmain.html

[3] https://mpi-inf.mpg.de/departments/databases-and-information-systems/research/neural-ir/cast19

[4] https://trec.nist.gov/data/webmain.html

Run	MAP	MRR	nDCG@3
mpi_base	0.173	0.508	0.234
mpi_bert	0.166	0.597	0.319
ug_cedr_rerank [2]	0.216	0.643	0.356

Table 1: Selected results from the CAsT 2019 overview paper. [2]

CEDR-KNRM outperforms BERT-KNRM, so CEDR's inclusion of BERT's CLS token may be improving performance; and *(2)* CEDR-KNRM used a different first stage retrieval method, which may have better recall; and *(3)* different training data was used. Given these differences, it is difficult to assess the impact of our first stage query expansion and our added attention module. The inclusion of BERT's CLS token does outweigh the combined impact of these differences, though we remark that all three differences may not have contributed positively to `mpi_bert`'s performance. We leave experiments assessing the impact of each individual change with respect to CEDR-KNRM for future work.

References

[1] Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)* (2013).

[2] Dalton, J., Xiong, C., and Callan, J. Cast 2019: The conversational assistance track overview. In *Proceedings of The Twenty-Eight Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019* (2019).

[3] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] MacAvaney, S., Yates, A., Cohan, A., and Goharian, N. Cedr: Contextualized embeddings for document ranking. In *SIGIR* (2019).

[5] Nogueira, R., and Cho, K. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085* (2019).

[6] Robyn Speer, J. C., and Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of AAAI31* (2017), vol. 1, pp. –.

[7] Xiangyang Zhou, Lu Li, D. D. Y. L. Y. C. W. X. Z. D. Y., and Wu, H. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), vol. 1, pp. –.

[8] Xiong, C., and Callan, J. Query expansion with freebase. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval* (New York, NY, USA, 2015), ICTIR '15, ACM, pp. 111–120.

[9] Xiong, C., Dai, Z., Callan, J., Liu, Z., and Power, R. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research & Development in Information Retrieval* (2017), ACM.

[10] Yang, P., Fang, H., and Lin, J. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2017), SIGIR '17, ACM, pp. 1253–1256.