Wilfrid Laurier University at the NIST TREC 2020: Conversational Assistance Track

Max Niebergall[†]
Department of Physics and Computer Science
Wilfrid Laurier University
Waterloo, Ontario, Canada
Max@MaxNiebergall.com

Jiashu Zhao[††]
Department of Physics and Computer Science
Wilfrid Laurier University
Waterloo, Ontario, Canada
JZhao@WLU.ca

1. INTRODUCTION

For TREC 2020, we submitted two runs to the Conversational Assistance Track (CAsT):

WLU_ManUttOnly, for this run we used indri search to retrieve 1000 candidate results for each query, followed by reranking with a hierarchical session-based learning model with BERT encoding, and evaluated on the manually rewritten conversational query set.

WLU_RawUttOnly, this run is the same as above except we evaluated on the raw conversational query set without any rewriting.

This report details the system we designed to generate these runs, as well as an evaluation of each run.

Overall, this system consists of four broad steps:

1. The newest utterance is used to retrieve 1000 sample answer paragraphs from the dataset using the Indri document search engine.
2. The sequences of utterances and their respective sample answer paragraphs the are then both encoded using a pretrained BERT-768 model [1].
3. The sequences of encoded utterances and encoded sample answer paragraphs are used as input to the main deep neural network, which determines a score in the range *[0,1]*.
4. Finally, the results are reranked according their scores.

We designed the approach in this paper based on one of our previous work, the hierarchical deep learning model DPHA [2]. We improved the model to incorporate the background information provided by the pretrained BERT model, in order to make up for the size of the dataset. Additionally, we leveraged a part of the MS MARCO [3] search dataset in training our model, though a larger portion of MS MARCO could have been used. Our work is also inspired by several work from CAsT 2019 [5], [6], [7].

2 METHODOLOGY

In this section we describe the details of our hierarchical session-based learning model, including an overview of the dataset, required data preprocessing, our document retrieval model, our choice of document encoding system, and our document reranking model.

2.1 Data Set and Data Preprocessing

The first step is to transform the dataset such that each query contains all the information required to predict a score for a potential answer document. For CAsT 2020, two sets of data were provided. Firstly, there is the dataset used for CAsT 2019, which contains 30 example sequences (called topics) each in the form of a sequence of utterances (i.e. questions) with some meta datía. Each of these utterances is associated with a number of (document ID, score) pairs which represent the training labels. For CAsT 2020, the evaluation utterances are associated with a canonical result document ID, as well as some rewritten versions of each utterance. Our model requires that each query contain the current utterance, and a sequence of previous utterances. For training, each query also needs to be associated with a potential

[†] Max Niebergall worked on this project as part of the Undergraduate Student Research Award at WLU and is now attending the MDSAI program at the University of Waterloo.

[††] Jiashu Zhao supervised this project and is an Assistant Professor at WLU.

answer document and its score. In the data preprocessing stage, a script reads the datasets, searches for potentially relevant documents (for evaluation only), performs document embedding, and outputs a series of files each containing: the topic number, turn number, canonical score (for training only), and encoded versions of the new utterance, past utterances, potential answer document, and the canonical response document (for evaluation only).

2.2 Document Retrieval

Because the set of available documents is so large, a conventional search engine such as Indri [8] is more convenient to produce a list of potential documents, since doing so with our deep learning model would be prohibitively expensive. Thus, we propose a document retrieval step which is separate from document encoding and document processing, and which can retrieve documents from a big set with sufficient speed. During preprocessing, we use a simple and naïve document retrieval process which uses the text of the newest utterance in Indri's #combine tag as the query for a default configuration of version 5.17 of the Indri document search engine with Krovetz stemming from which 1000 thousand seemingly relevant documents are retrieved. These documents are the encoded as described in the next section, before being reranked by our document scoring model as described in section 2.4.

2.3 Document Encoding

Since the dataset of labeled documents is relatively small, we chose to incorporate background information on the English language by using a pretrained language model, to decrease the amount of data needed to train our model to extract useful information from the text. To that end, we used a pretrained BERT-768 model to encode each utterance. In practice, we used BERT-as-a-service [4] to easily encode our utterances and documents. Using BERT-as-a-service allowed us to encode all utterances and documents as part of the preprocessing step, instead of during model prediction or training. However, using this technique we were unable to finetune the BERT model on our training set, which limits the effectiveness of the BERT model at this task.

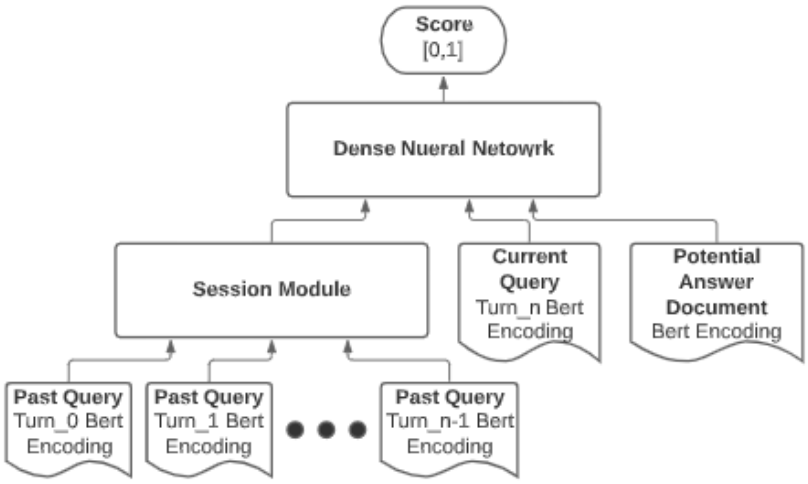

Figure 1: The architecture of our predictive model

2.4 Document Scoring Model

Figure 1 shows the general architecture of our proposed predictive model. The past utterances, current utterance, and a potential answer document are encoded as explained in Section 2.3. Information from multiple past turns is learned in the Session Module, in which we adopt our previous work, DPHA model, proposed in [2]. The model structure is shown in Figure 2. It models dynamic session intent from the preceding queries by applying multi-head self-attention with dropout to the sequence of inputs before applying GRUs to encode the session by learning sequential information in the session forwardly. Then, a soft attention layer is applied to generate the session's annotation by aggregating the hidden states of the GRUs.

Finally, the session module's annotation of the session, the encoded query, and the encoded potential answer document are fed into a dense layer using sigmoid activations which computes the final score of the potential answer document. In training, the score assigned to the potential answer document by TREC is compared to the predicted score and binary cross entropy loss is computed.

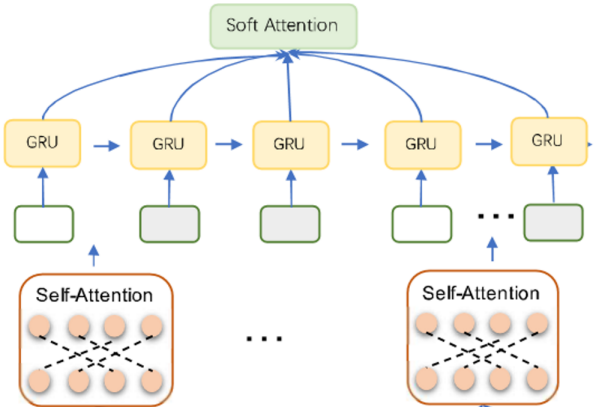

Figure 2: The Session module from the DPHA framework

From: A Dynamic Product aware Learning Model for E-commerce Query Intent Understanding [2].

3 EVALUATION OF RESULTS

TREC lists four measures as the most important for this track. They are: NDCG@3, NDCG@5, NDCG@1000, and AP@1000 where NDCG is the Normalized Discounted Cumulative Gain over the top number of results, and AP@1000 is the Average Precision over the entire result set returned for each turn. For this project, NDCG can be thought of as a measure of the effectiveness of our scoring model given our document retrieval model, while AP@1000 is a measure of the effectiveness of our document retrieval model. In the table below, we present the average value of these measures. Note: out of 208 questions and the 208'000 results our model returned only around 10'000 were judged so these scores should be taken as an estimate only.

Table 1: Main measures of results by run

Average Score	**ManUttOnly**	**RawUttOnly**	Mean Median Raw Run
NDCG@3	0.0665	0.0222	0.2796
NDCG@5	0.0695	0.0215	0.2735
NDCG@1000	0.2872	0.1106	0.3749
AP@1000	0.0456	0.0147	0.1801

Table 1 shows that the Manually rewritten queries return results that are about 3 times as relevant as the Raw utterances alone, by these measures. This indicates that our model is not as effective at interpreting raw utterances as the human rewriters. Therefore, raw utterance interpretation is an area of potential for improvement in future iterations of this system. When comparing the RawUttOnly results to the mean across all turns of the median performing raw runs submitted in this track, we see that the median runs outperform our model in these measures. The difference indicates that our document retrieval system is ripe for improvement.

4 FUTURE WORK

After analyzing the performance results, we believe there are several areas of our project in which future improvements could lead to big improvements in overall results.

Firstly, our document retrieval process is simple, and our AP@1000 scores show that it limits the effectiveness of the rest of our system. Potential improvements to the document retrieval process include:

- Including information from past turns in the query to the document search engine
- Performing query rewriting before sending the query to the document search engine
- Performing keyword and named entity identification and extraction prior to searching
- Evaluating alternative document search engine algorithms for this problem

Secondly, our document encoding process can be further improved. Currently, it uses only the general BERT-768 model, without finetuning which leaves great room for improvement. This year alone, several alternative language models have shown to be even more powerful than BERT and exploring those could improve the effectiveness of our embeddings. Using a language model which is finetuned would likely improve effectiveness of our document scoring model and could greatly improve our NDCG scores. At the same time, it is unlikely that the CAsT dataset alone would be of sufficient size to fine tune BERT on, and we would need to bring in external datasets like MS MARCO for this task to be effective. Finetuning BERT and augmenting out dataset with more MS MARCO datapoints (as well as any other text retrieval datasets that may be developed in the future) are both areas with great potential in future research.

ACKNOWLEDGMENTS

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). We also thank Wilfrid Laurier University Faculty of Science for funding this project as a part of the Faculty start-up fund and Undergraduate Student Research Award.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805v2

[2] Jiashu Zhao and Hongshen Chen, Dawei Yin. 2019. A Dynamic Product aware Learning Model for E-commerce Query Intent Understanding. In Proceedings of The 28th ACM International Conference on Information and Knowledge Management (CIKM '19). ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3358055

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. https://arxiv.org/abs/1611.09268

[4] Han Xiao. bert-as-service. 2018. https://github.com/hanxiao/bert-as-service.

[5] Jeffrey Dalton, Chenyan Xiong, Jamie Callan. TREC CAsT 2019: The Conversational Assistance Track Overview. 2020. https://arxiv.org/abs/2003.13624

[6] Charles L. A. Clarke. WaterlooClarke at the TREC 2019 Conversational Assistant Track. 2019. https://trec.nist.gov/pubs/trec28/papers/WaterlooClarke.C.pdf

[7] Jheng-Hong Yang, Sheng-Chieh Lin, Jimmy Lin, Ming-Feng Tsai, Chuan-Ju Wang. Query and Answer Expansion from Conversation History. 2019. https://trec.nist.gov/pubs/trec28/papers/CFDA_CLIP.C.pdf

[8] The Lemur Project. Indri Document Search Engine version 5.17. 2002. https://www.lemurproject.org/indri.php