

# UMass at TREC 2019 Conversational Assistance Track

Helia Hashemi and W. Bruce Croft

Center for Intelligent Information Retrieval  
College of information and Computer Science  
University of Massachusetts Amherst  
Amherst, MA 01003  
{hhashemi,croft}@cs.umass.edu

## 1 Introduction

This is an overview of University of Massachusetts efforts in providing passage retrieval run submissions for the TREC 2019 Conversational Assistance Track (CAsT). We adopted recent neural approaches for the task. The goal is to retrieve passages that are different from what traditional methods retrieve, in order to enrich the candidates pool.

## 2 Method

We formalize the TREC CAsT problem as follows. We are given an information seeking conversation dataset  $D = \{(U_i, R_i, Y_i)\}_{i=1}^N$ , where  $U_i = \{u_i^1, u_i^2, \dots, u_i^{t-1}, u_i^t\}$  in which  $\{u_i^1, u_i^2, \dots, u_i^{t-1}\}$  is the dialog context and  $u_i^t$  is the input utterance in the  $t$ -th turn.  $R_i$  and  $Y_i$  are a set of response candidates  $\{r_i^1, r_i^2, \dots, r_i^k\}_{k=1}^M$ , and the corresponding binary labels  $\{y_i^1, y_i^2, \dots, y_i^k\}$ , where  $y_i^k = 1$  means  $r_i^k$  is a relevant response to  $U_i$ . For any given  $U_i$ , the goal is to learn a ranking model  $f(\cdot)$  to retrieve relevant responses.

In our submissions, the candidate responses and dialog contexts are modeled by a deep neural matching network. We used the approach recently proposed by Yang et al. [5] in our submissions. Given a candidate response  $r_k$  and an utterance  $u_i^t$  in the context  $U_i$ , the model firstly looks up a global embedding dictionary to represent  $r_k$  and  $u_i^t$  as two sequences of embedding vectors  $E(r_i^k) = [e_{r,1}, e_{r,2}, \dots, e_{r,l_r}]$  and  $E(u_i^t) = [e_{u,1}, e_{u,2}, \dots, e_{u,l_u}]$ , where  $e_{r,i} \in \mathbb{R}^d$  and  $e_{u,i} \in \mathbb{R}^d$  are the embedding vectors of the  $i$ -th word in  $r_i^k$  and  $u_i^t$ , respectively. Here, we adopt an interaction-focused method to learn the matching patterns. Specifically, with  $E(r_i^k) \in \mathbb{R}^{d \times l_r}$  and  $E(u_i^t) \in \mathbb{R}^{d \times l_u}$ , we build two interaction matrices  $M_1$  and  $M_2$ , such that  $M_1$  is a word pairwise similarity matrix and  $M_2$  is a sequence hidden representation similarity matrix. Later, these two matrices will be the two input channels of a convolutional neural network (CNN) to learn important matching features.

To be more precise,  $M_1$  models the word pairwise similarity between  $r_i^k$  and  $u_i^t$  via the dot product similarity between the embedding representations. On the

other hand, for making the matrix  $M_2$ , we use BiGRU [1] to obtain the hidden representation of the whole text for both  $u_i^t$  and  $r_i^k$ . Each cell of  $M_2$  will be the similarity of different parts of these two hidden representation.

These matrices are fed to a CNN followed by max pooling, and then a BiGRU layer to model the dependency and temporal relationship of utterances in the conversation according the previous works [4]. We finally concatenate the output hidden states, and feed the learned representation to a multi-layer perceptron (MLP) network to generate the matching score.

We train the model in the pairwise setting, and for loss function we use a pairwise hinge loss. The parameters of the deep matching network are optimized using back-propagation with Adam optimizer [2].

For more information on the model design, refer to [5].

### 3 Experimental Setup

We used the *TREC-CAsT Tools* for processing and parsing all the collections, including MS MARCO Passage Ranking collection, TREC CAR paragraph collection v2.0, and TREC Washington Post Corpus version 2. We also cleaned the collection text by omitting HTML tags.

We train our model on MS MARCO Conversational Search Sessions. Each session starts with one utterance at the beginning, then, as the conversation goes further the conversation history increases. MS MARCO Conversational Search Session data has single turn relevance judgement annotation. We use these judgements as weak labels, since it enables us to train the model on large amount of the data provided by Microsoft. Therefore, the label of each passage for any sequence of utterances depends on label of that passage for the last utterance. By doing so, we were able to train our model on 1091553 instances. For negative samples of the training set, we use, on average, the top 10 retrieved passages provided in the MS MARCO collection.

The maximum length of utterances and each responses was set to 90 and 150, respectively. The initial learning rate was set to 0.001. The parameters of Adam,  $\beta_1$  and  $\beta_2$  were set to 0.9 and 0.999, respectively. The dropout rate is 0.6. We set the window size of the convolution and pooling kernels as (3,3). The number of convolution kernels was 2. The dimension of the hidden states of BiGRU layer was set to 100. The maximum conversation context length was 13. We padded short contexts with zero. For the pre-trained embeddings used in our experiments, we picked the vectors learned by the Word2Vec tool [3] with the Skip-gram model. The max skip length between words and the number of negative examples is set as 5 and 10, respectively. The dimension of word vectors is 200. Word embeddings are updated during the training process.

**UMass Submissions:** Finally, we re-rank the top 300 passages retrieved by the Query Likelihood model using Indri. We used the Indri runs provided by the track organizers. For co-reference resolution, the AllenNLP toolkit was used. Stopwords were removed using the Indri stopword list. Two different runs were submitted to the TREC CAsT track this year, which are as follows:

- The first run is a result list of the described model computed based on the original file of the Evaluation topics year 1 V1.0.
- The Second run is a result list of the described model computed based on the Resolved Topic Annotations.

## 4 Acknowledgements

We would like to thank Liu Yang for open-sourcing the deep matching model [5] and for his invaluable comments on the model.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1715095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

## References

1. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR (2014)
2. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR (2014)
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
4. Wu, Y., Wu, W., Zhou, M., Li, Z.: Sequential match network: A new architecture for multi-turn response selection in retrieval-based chatbots. CoRR (2016)
5. Yang, L., Qiu, M., Qu, C., Guo, J., Zhang, Y., Croft, W.B., Huang, J., Chen, H.: Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. CoRR (2018)