UCAS at TREC-2019 Deep Learning Track

Xuanang Chen[1,2], Canjia Li[1,2], Ben He[1,2], and Yingfei Sun[1]

[1]University of Chinese Academy of Sciences
[2]CIP Laboratory, Institute of Software, Chinese Academy of Sciences
{chenxuanang19, licanjia17}@mails.ucas.ac.cn, {benhe, yfsun}@ucas.ac.cn

Abstract. This paper describes the experiment conducted for our participation in the TREC-2019 Deep Learning track [1]. We test the effectiveness of two pre-trained language models, BERT [2] and XLNet [3], for the re-ranking subtask of the document ranking task, with an adoption of the passage-level document ranking approach as proposed in [4]. Our preliminary results indicate that the uses of BERT and XLNet lead to comparable performance.

1 Introduction

The UCAS participation in the TREC 2019 Deep Learning track (DL2019) aims to study how to learn neural IR models out of large-scale training data. Specifically, a large set of human-generated training labels from the MS-MARCO dataset are used. In our experiments, we only focus on the re-ranking subtask of the document ranking task. Recently, neural models pre-trained on a language modeling task, such as BERT and XLNet, have advanced the state-of-the-art results on several ranking tasks [4–8]. Therefore, we re-purpose BERT and XLNet as document re-rankers to score or rank the candidate documents, in order to achieve competitive results.

The rest of the paper is organized as follows. Section 2 gives a detailed introduction to the approach used in our experiments. Section 3 presents the experimental settings, results. Finally, Section 4 concludes our experiments.

2 Method

In this section, we give a detailed introduction to our approach for the document re-ranking subtask.

2.1 Document Split

Due to the maximum sequence length limitation of BERT and XLNet, we adopt a recent passage level approach for document retrieval introduced in [4]. Documents are split into overlapping passages (up to 30 passages with 150 words for a document), and the title, if available, is added to the beginning of every passage. Simply, we consider all passages from a relevant document as being relevant, and passages from a irrelevant document as being irrelevant. The neural re-ranker predicts the relevance of each passage with a query independently, and the document score is given by the score of the best passage (MaxP) or the sum of all passage scores (SumP). In our experiments, the re-ranker with MaxP performs better.

2.2 Re-ranker with BERT

In BERT re-ranker, we use the BERT-Large model (bert-large-uncased) to re-rank the top-100 documents. We truncate the query to have at most 64 tokens and truncate the passage text such that the concatenation of query, passage, and separator tokens have the maximum length of 256 tokens. The input format of BERT re-ranker is [CLS] [query] [SEP] [passage] [SEP], and the final hidden vector of the [CLS] token is used as input to a single layer neural network to obtain the probability (score) of the passage being relevant. After that, we produce the score of a document according to the scores of its split passages, and eventually, all candidate documents are re-ranked by the document scores received.

2.3 Re-ranker with XLNet

In XLNet re-ranker, we use the XLNet-Large model (xlnet-large-cased) to re-rank the top-100 documents. We only truncate the passage text such that the concatenation of query, passage, and separator tokens have the maximum length of 256 tokens. Different from BERT re-ranker, the input format of XLNet re-ranker is [query] [SEP] [passage] [SEP] [CLS]. Then, akin to the BERT re-ranker, we use the final hidden vector of the [CLS] token to re-rank all candidate documents.

We start training from a pre-trained BERT or XLNet model, and fine-tune them to our re-ranking subtask using the cross-entropy loss [5]:

$$L = -\sum_{j \in J_{\text{pos}}} \log(s_j) - \sum_{j \in J_{\text{neg}}} \log(1 - s_j) \tag{1}$$

where J_{pos} is the set of indexes of the relevant passages and J_{neg} is the set of indexes of non-relevant passages split from top-100 documents provided.

3 Experimental Setting and Results

3.1 Experimental Setup

The corpus has 3.2 million documents, up to about 22GB. The training dataset has 367,013 queries, the validation dataset has 5,193 queries, and the test dataset has 200 queries. Before experiment, we need to generate train triples and query-passage pairs.

We use the top-100 result file in train dataset to generate the train triples. After splitting the documents, we preserve all positive passages in relevant documents and sample a negative passage in irrelevant documents for a query. Then we get about 4,343k train triples for model fine-tuning. We also use the top-100 results in validation and test dataset to generate query-passage pairs of validation and evaluation. Here, we preserve all passages in documents and get about 350k pairs for model evaluation.

We fine-tune the model using TPUs on Google Cloud[1] with a batch size of 32 for both BERT and XLNet. The model is trained on the above train triples (about 8GB) for 400k iterations, and the model with the best MRR@10 metric on validation queries is chosen, and evaluated on test queries.

3.2 Results

We submitted three runs for the document re-ranking subtask, the differences among the three runs are summarized in Table 1 (PLM denotes the pre-trained language model, and Aggregation means the way to get the score of document from the scores of passages.):

Table 1. Run submission summary

RunID	PLM	Aggregation	Train Steps
ucas_runid1	BERT (bert-large-uncased)	MaxP	370k
ucas_runid2	XLNet (xlnet-large-cased)	MaxP	360k
ucas_runid3	BERT (bert-large-uncased)	MaxP	375k

Table 2. Evaluation results

runID	RR(MS)	RR	NDCG@10	AP
ucas_runid1	**0.442**	0.911	**0.644**	0.264
ucas_runid2	0.431	**0.950**	0.635	0.253
ucas_runid3	0.435	0.899	0.642	**0.268**

The evaluation results of our runs for document re-ranking are shown in Table 2. The best values are highlighted in boldface. From the results above, we find no apparent winner between the uses of BERT and XLNet. A noticeable finding is that the run ucas_runid2 based on XLNet has the best RR, but does not outperform the BERT re-rankers for other evaluation metrics.

4 Conclusions

In this paper, we describe the system based on BERT and XLNet for document re-ranking subtask of TREC-2019 Deep Learning track. Our pilot experiments

[1] https://cloud.google.com/tpu

show no significant difference between the uses of BERT and XLNet within the passage-level document ranking approach as in [4]. A likely cause is the missing next passage prediction (NSP) component for pre-training the cased XLNet-large used in our experiments.

Acknowledgments

This work is supported by the University of Chinese Academy of Sciences.

References

1. N. Craswell, B. Mitra, E. Yilmaz, and D. Campos, "Overview of the trec 2019 deep learning track," in *TREC (to appear)*, 2019.
2. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.
3. Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *CoRR*, vol. abs/1906.08237, 2019.
4. Z. Dai and J. Callan, "Deeper text understanding for IR with contextual neural language modeling," *CoRR*, vol. abs/1905.09217, 2019.
5. R. Nogueira and K. Cho, "Passage re-ranking with BERT," *CoRR*, vol. abs/1901.04085, 2019.
6. W. Yang, H. Zhang, and J. Lin, "Simple applications of BERT for ad hoc document retrieval," *CoRR*, vol. abs/1903.10972, 2019.
7. Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, "Understanding the behaviors of BERT in ranking," *CoRR*, vol. abs/1904.07531, 2019.
8. H. Padigela, H. Zamani, and W. B. Croft, "Investigating the successes and failures of BERT for passage re-ranking," *CoRR*, vol. abs/1905.01758, 2019.