ICIP at TREC-2020 Deep Learning Track

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

[1] University of Chinese Academy of Sciences, Beijing, China
[2] Laboraroty of Chinese Information Processing, Institute of Software, Chinese Academy of Sciences, Beijing, China
chenxuanang19@mails.ucas.ac.cn
{benhe, yfsun}@ucas.ac.cn, sunle@iscas.ac.cn

Abstract. This paper describes the ICIP participation in TREC 2020 Deep Learning Track. We apply BERT [1] to the re-ranking subtask of the document ranking task, with an adoption of the passage-level BERT re-ranker [2]. We utilize both the passage and document ranking dataset for model training, and the noisy training samples in generated document training set will be filtered, to guarantee and boost the ranking effectiveness. Additionally, we also distill smaller BERT models, on top of the recent knowledge distillation (KD) method on BERT, called Simplified TinyBERT [3], to investigate the influence of KD on the document ranking task.

1 Introduction

The ICIP participation in the TREC 2020 Deep Learning track aims to study how to learn neural IR models out of the large-scale training data, and how to improve the effectiveness of BERT-based ranking models, and additionally how to distill the large and expensive BERT-based ranking model to the small, efficient and effective one. In our experiments, we only focus on the re-ranking subtask of the document ranking task.

Recently, contextual pre-trained language model, like BERT [1], has advanced the state-of-the-art results on several ranking tasks [2, 4–7]. Besides, the Knowledge Distillation (KD) technique [8] has been applied to compress BERT model for fast inference while maintaining the comparable performance on multiple NLP tasks [3, 9–14]. Therefore, we utilize BERT model as the document re-ranker to score and re-rank the candidate documents, and meanwhile apply the KD method to the BERT re-ranker, to investigate the effectiveness of distilled BERT models on the document ranking task.

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, models and ranking 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 and Passage Filter

Due to the maximum sequence length limitation of BERT, all documents are split into up to 30 overlapping passages (150 whole words and 75 overlapping words), and the title is added to the beginning of every passage if it is available. Simply, for the training set, the passages from a relevant document can be considered as relevant, and the passages from a irrelevant document as irrelevant [2]. But this simple process may produce a number of noisy training examples, which will negatively affect training a robust model.

Different from this simple process in previous work [2, 15], we add a passage filter step to clean the training samples generated by the document split step, as the document is labeled relevant if it contains a relevant passage (paragraph or chunk) in the MS MARCO document dataset. In the passage filter step, we use a passage-level BERT re-ranker to filter out the irrelevant passages split from relevant documents for a query. More specifically, we only preserve the top-ranked passages in a relevant document according the relevance scores of passages given by a BERT ranker fine-tuned on passage ranking dataset. But note that we still preserve all overlapping passages for the validation and test set, to really judge the relevance of a query-document pair.

2.2 Passage-level BERT Re-ranker

For the passage-level BERT re-ranker [2], we adopt the BERT-Large model (bert-large-uncased, 24 layers), which also behaves as the teacher model in the KD training procedure where its ranking knowledge will be distilled to the smaller BERT model (12 layers). The passage text could be truncated such that the concatenation of query, passage, and separator tokens has the maximum length of 256 tokens. The input format of BERT re-ranker is [CLS] [query] [SEP] [passage] [SEP], and the final pooled hidden vector of the [CLS] token is fed into a single layer feed-forward network to obtain the probability (score) of the passage being relevant. We adopt the same settings in [4] to fine-tune BERT model to the document re-ranking task.

The BERT re-ranker predicts the relevance of each passage with a query independently. After that, we generate the score of a document according to the scores of its split passages, namely, the max score of the passages (*MaxP*) or the average score of the top-K passages (*K-Max-AvgP*). Eventually, all candidate documents are re-ranked by the document scores received. In our experiments, K is set as 2, and we demonstrate the better effectiveness of the *2-Max-AvgP* compared to *MaxP*.

2.3 Knowledge Distillation on BERT

Due to the expensive computation cost of BERT during inference, some knowledge distillation methods on BERT have been proposed, such as DistilBERT [10], BERT-PKD [11], TinyBERT [9], MobileBERT [12] and MiniLM [14], to distill a large BERT model (teacher) into a smaller BERT model (student), which not only has a faster inference speed but also maintains the comparable performance.

In our experiments, we adopt the Simplified TinyBERT method [3] to distill smaller BERT re-ranker. Simplified TinyBERT simultaneously distills the representation of embedding and hidden states, the attention behavior and the probability output (soft label) of the teacher BERT model, and also adds hard label (supervision signals from the training examples) to further boost the document ranking performance of smaller BERT models. We refer the readers to the paper [3] for further details of the KD procedure of Simplified TinyBERT.

3 Experiments

In this section, the implementation details of our experiments are described, followed by the ranking performance of our models.

3.1 Experimental Setup

The document corpus contains about 3.2 million documents, 367,013 training queries, 5,193 validation queries, and 200 queries for test. We use the official top-100 result file of queries to generate train triples, validation and test pairs. For the passage corpus, we choose the *Train Triples Small* as the training samples, and there is no need to apply the split and filter step here.

In generating the training triples, for a query, after splitting the top-100 documents, we sample a negative passage in the passage pool of irrelevant documents for every passage in the relevant document. Based on this preliminary training triples (query, positive passage, negative passage, about 4,343k), we utilize a passage filter (BERT-Large model released in [4]) to remove the training triples which contains a fake positive passage, and only **retain the training triples which contains the five top-ranked positive passages**. Eventually, the total training data contains about 1,646k training triples for model fine-tuning and distillation. But for query-passage pairs in validation and evaluation sets, we preserve all passages of a document, in order to guarantee getting the real performance of the BERT re-ranker.

We carry out our experiments on three TITAN RTX 24G GPUs with Mixed Precision Training [16]. We use Adam optimizer with a weight decay of 0.01 with a learning rate 1e-06 for fine-tuning and a learning rate 5e-05 for distillation. Models are fine-tuned and distilled with batch size of 32 and 64, respectively. The model with the best MRR@10 metric on validation set is chosen, and evaluated on test sets.

Table 1. The summary of models. **Aggregation** refers the way to get the score of a document according to the scores of its split passages during **inference**, *MaxP* means the max score of passages and *2-Max-AvgP* means the average score of the top-2 ranked passages.

Model ID	Model (Config)	Training Dataset	Distillation	Aggregation
1	BERT_Large (L24_H1024_A16)	Passage	-	*MaxP*
2				*2-Max-AvgP*
3	BERT_Large (L24_H1024_A16)	Passage & Document	-	*MaxP*
4				*2-Max-AvgP*
5	BERT_Distilled (L12_H1024_A16)	Document	One Step	*MaxP*
6			One Step + Hard Label	

Table 2. Evaluation results on TREC 2019 & 2020 DL test queries in document re-ranking subtask, which contains 43 and 45 queries, respectively. Statistical significance at p-value < 0.05 is marked with Model ID for comparisons to each model.

Model ID	TREC 2019 DL Test			TREC 2020 DL Test		
	MRR	NDCG@10	MAP	MRR	NDCG@10	MAP
1	0.9225^{2}	0.6793^{2}	0.2877^{6}	0.9444	0.6440	0.4274^{2}
2 (ICIP_run3)	$\mathbf{0.9729}^{1}$	$\mathbf{0.7086}^{1}$	$\mathbf{0.2907}^{4,6}$	**0.9667**	0.6528	$0.4360^{1,5,6}$
3 (ICIP_run1)	0.9554	0.6886	0.2815	0.9630	$0.6623^{5,6}$	0.4333
4	0.9496	0.6857	0.2752^{2}	**0.9667**	$\mathbf{0.6685}^{5,6}$	$\mathbf{0.4389}^{5,6}$
5 (ICIP_run2)	0.9535	0.6823	0.2785	0.9407	$0.6322^{3,4}$	$0.4206^{2,4}$
6	0.9477	0.6871	$0.2734^{1,2}$	0.9333	$0.6353^{3,4}$	$0.4192^{2,4}$

3.2 Models and Results

The details of different models are summarized in Table 1, which contains the models that produced our three submitted runs. **Model 1** is the BERT-Large model fine-tuned on passage ranking dataset (*Train Triples Small*), adopting *MaxP* aggregation way to get the score of a document from the scores of its split passages. **Model 2** is the same as **Model 1**, but it adopts *2-Max-AvgP* aggregation way, and also produces our submission ICIP_run3. **Model 3** and **Model 4** are first fine-tuned on passage ranking dataset, then fine-tuned on document ranking dataset (training triples generated as in Section 2.1), adopting *MaxP* and *2-Max-AvgP* aggregation way, respectively. **Model 5** and **Model 6** are 12-layer models distilled from the BERT-Large model (**Model 3** or **Model 4**) in the Simplified TinyBERT distillation setting on document ranking dataset, adopting *MaxP* aggregation way. **Model 3** and **Model 5** produce our submission ICIP_run1 and ICIP_run2, respectively.

The evaluation results of our models for document re-ranking are shown in Table 2. For a more comprehensive comparison, we present the evaluation results on both TREC 2019 and 2020 DL Test set. The best values are highlighted in boldface and statistical significance for paired two-tailed t-test is reported. From the results above, we find that **Model 2** outperforms other models on TREC

DL 2019 Test set, and meanwhile **Model 4** behaves better than other models on TREC DL 2020 Test set.

There is a lit difference on the effectiveness of *2-Max-AvgP* aggregation way on TREC 2019 and 2020 DL Test set, compared to *MaxP* aggregation way. From Table 2, on TREC 2019 DL Test set, we can see that **Model 2** significantly outperforms **Model 1** in terms of MRR and NDCG@10, but *2-Max-AvgP* aggregation way fails to improve **Model 4**; meanwhile, *2-Max-AvgP* aggregation way behaves better than *MaxP* aggregation way on TREC 2020 DL Test set (**Model 1** vs. **Model 2**, and **Model 3** vs. **Model 4**). There is still a bit large of margin on the ranking performance between distilled models (**Model 5** and **Model 6**) and the teacher model (**Model 3**) on TREC 2020 DL Test set. The two distilled models (**Model 5** and **Model 6**) behave similarly, and adding hard label could boost the NDCG@10 metric on both test sets.

4 Conclusions

In this paper, we describe the system based on BERT model for the document re-ranking subtask in TREC 2020 Deep Learning track. In the passage-level BERT ranker setting, the BERT ranker fine-tuned on passage ranking dataset can be transferred to the document ranking task effectively. The superiority of *K-Max-AvgP* aggregation way seems to be related to the model settings and test data, compared to the *MaxP* aggregation way. Distilled BERT models do not behave as well as in [3], may mainly because of the different setting of the teacher model. We plan to investigate these issues in future research.

Acknowledgements

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

References

1. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1). pp. 4171–4186. Association for Computational Linguistics (2019)
2. Dai, Z., Callan, J.: Deeper text understanding for IR with contextual neural language modeling. In: SIGIR. pp. 985–988. ACM (2019)
3. Chen, X., He, B., Hui, K., Sun, L., Sun, Y.: Simplified tinybert: Knowledge distillation for document retrieval. CoRR **abs/2009.07531** (2020)
4. Nogueira, R., Cho, K.: Passage re-ranking with BERT. CoRR **abs/1901.04085** (2019)
5. Padigela, H., Zamani, H., Croft, W.B.: Investigating the successes and failures of BERT for passage re-ranking. CoRR **abs/1905.01758** (2019)
6. Qiao, Y., Xiong, C., Liu, Z., Liu, Z.: Understanding the behaviors of BERT in ranking. CoRR **abs/1904.07531** (2019)
7. Yang, W., Zhang, H., Lin, J.: Simple applications of BERT for ad hoc document retrieval. CoRR **abs/1903.10972** (2019)

8. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR **abs/1503.02531** (2015)
9. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q.: Tinybert: Distilling BERT for natural language understanding. CoRR **abs/1909.10351** (2019)
10. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR **abs/1910.01108** (2019)
11. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression. In: EMNLP/IJCNLP (1). pp. 4322–4331. Association for Computational Linguistics (2019)
12. Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., Zhou, D.: Mobilebert: a compact task-agnostic BERT for resource-limited devices. CoRR **abs/2004.02984** (2020)
13. Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., Lin, J.: Distilling task-specific knowledge from BERT into simple neural networks. CoRR **abs/1903.12136** (2019)
14. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. CoRR **abs/2002.10957** (2020)
15. Chen, X., Li, C., He, B., Sun, Y.: UCAS at TREC-2019 deep learning track. In: TREC. NIST Special Publication, vol. 1250. National Institute of Standards and Technology (NIST) (2019)
16. Micikevicius, P., Narang, S., Alben, J., Diamos, G.F., Elsen, E., García, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., Wu, H.: Mixed precision training. In: ICLR (Poster). OpenReview.net (2018)