A Multiple Models Ensembling Method in Trec Deep Learning

Liyu Cao[1,*], Yixuan Qiao[1,*], Hao Chen[1], Peng Gao[1], Yuan Ni[1], and GuoTong Xie[1]

[1]Ping An Technology (Shenzhen) Co., Ltd.,
{caoliyu922, qiaoyixuan528, chenhao305, niyuan422, xieguotong}@pingan.com.cn

Abstract

This paper is to describe our participation in the passage ranking tasks of TREC 2020 Depp Learning Track. We take leverage of several pre-trained models, such as BERT, ELECTRA and XLNET, to re-rank passages. Firstly, we use corpus in this track and enhanced next sentence prediction task to pretreain a new BERT large model. Secondly, we fine-tune the new BERT model as well as ELECTRA, XLNET and ALBERT with selected triplet data. Then, we ensemble several different kind of models to score and rank top-1000 passage. Finally, we select and re-rank top-10 passages in the former step according to the similarity to the top-1 passage to reduce noise.

1 Introduction

The Deep Learning Track continues to have the same tasks and goals as TREC 2019. Similar to the previous year, one of the main goals is to study what methods work best when a large amount of training data is available. It consists of two tasks: passage ranking and document ranking; and two subtasks in each case: full ranking and re-ranking. Each task is based on human-generated data, which comes from MS-MARCO dataset.

Our approach is mainly based on pre-trained models, such as BERT, ALBERT, ELECTRA and XLNET, which are state-of-art models in different natural language understanding tasks. Besides directly fine-tuning BERT with given training data, we try to retrain a new BERT model with part of whole training data. Based on the new models, then we fine-tune it with pairwise ranking strategy on the labeled data.

2 Our Approach

This section presents approaches we used in passage re-ranking task. About 1000 passages are selected by BM25 for each query, and we need to re-rank passages to ensure that top-10 passages are more relevant to the query. We use several pre-trained models to compare their performances in this task.

2.1 BERT-based Re-ranker

Pre-trained models, especially BERT[1], have been shown powerful in wide range of language understanding tasks. There are two steps: pre-training and fine-tuning. Firstly, we pre-train a new BERT model with a more difficult pre-training task, which is inspired from StructBERT[2] and IDST [3]. Secondly, we fine-tune the model parameters on training data with a pairwise ranking strategy to receive a score to represent the relevence between query and passage.

In the pre-training part, instead of using next sentence prediction as one of the pre-training objectives, we reinforce the task to previous sentence prediction and next sentence prediction. The original NSP task is too weak to train a powerful model, because it is easy to distinguish whether it is next sentence since they come from the same passage and have same topic. However, previous and next sentence prediction can take leverage of contextual information since it have to judge which sentence is former. Specifically,

given a sentence S1 from a passage, there is a one-third probability to choose the previous sentence, a one-third probability to choose the next sentence and a one-third probability to choose a random sentence from the other passage. Besides the new NSP task, we kept the MLM task as original and then train a new BERT model with them.

In the fine-tuning part, we follow the method [4] and treat the model as a classifier to get a relevent score. We pack one query and one passage with [SEP], then put the whole sentence to BERT model and use [CLS] vector in the last layer as a representation to compute a score. To take advantage of triplet training dataset, we use pairwise strategy to train BERT model. The triplet loss is:

$$L(q, p^+, p^-, \theta) = max(0, \theta - S(q, p^+) + S(q, p^-))) \quad (1)$$

where q represents query, p^+ represents positive passage and p^- represents negative passage, $S(q, p)$ represents score given by BERT model, $S = sigmoid(w \cdot h^L_{CLS})$, where w is a trainable parameter and h_{CLS} is the hiddent state of [CLS].

2.2 ALBERT-based Re-ranker

ALBERT is a lite BERT model which incorporates two parameter reduction techniques. [5] By factorized embedding parameterization, which means decomposing the large vocabulary embedding matrix into two small matrices, and cross-layer parameter sharing An ALBERT has 18x fewer parameters than BERT-large model and can be trained about 1.7x faster.

As we do in section 2.1, we use hidden vector of [CLS] in the last layer to re-rank all candidate passages. The loss function is the same as equation(1).

2.3 ELECTRA-based Re-ranker

ELECTRA proposes a more sample-efficient pre-training task called replaced token detection. [6] Firstly, we train a small generator network to replace some tokens with plausible alternatives. Then we train a discriminative model that predicts whether each token in the corrupted input was replaced by a generator sample or not.

we organize input as [CLS] query [SEP] passage [SEP] and feed them to ELECTRA and use the final hidden vector of [CLS] to get a relevant score. The loss function is also the same as equation(1).

2.4 XLNET-based Re-ranker

Different with BERT, XLNET is a generalized autoreggressive pre-training model[7]. In this re-ranker, we use XLNET-base model to re-rank top 1000 passages. We join query with positive passage or negative passage and sent them to XLNET model. The input format is [query][SEP][passage][SEP][CLS] and we use the final hidden vector of [CLS] to get a relevant score. The loss function is also the same as equation(1).

2.5 Re-rank top-10

After re-ranking, We find that top-10 passages sometimes contains irrelevant passages with query. In order to eliminate the influence of such negative passages, we assume that top-1 passage is a positive passage and other relevant passage should be similar to it.

We use glove-300[8] to compute vectors to represent words in a passage and then use the average of all these vectors as the passage's representation. By compute cosine similarity of the vectors, we can receive similarity of the passages with top-1 passage. We set a threshold to decide whether a passage's score, ranked from 2 to 10, should be modified. At last, we set the threshold as 0.1, which means if the max similarity score minus the minimum score is larger than 0.1, the minimum score should minus 0.2.

Table 1: overall performance of submitted models on test 2020

Run Tag	Group	Run Description	NDCG@10	mAP
pinganNLP-1	pinganNLP	BERT+ELETRA+XLNET ensemble (6 models)	0.7343	0.4896
pinganNLP-2	pinganNLP	pinganNLP-1 + re-rank top 10	0.7368	0.4881
pinganNLP-3	pinganNLP	only BERT ensemble (6 models) + re-rank top 10	0.7352	0.4918

3 Discussion and Conclusion

4 Experiment

4.1 Dataset

The corpus contains 1,010,916 queries on a collection of 8,841,823 passages. Sentences used in section 2.1 pre-training part are sampled from these passages. Triplet train dataset contains 397,756,691 pairs of query, positive and negative passages. However, it only consists of about 500,000 different queries. We randomly select one piece of data for each query, so the final triplet data we used contains 461,594 triplet pairs.

4.2 Result

Table 1 presents results from different models and ensembled models. As we can see an ensemble of BERT, ELECTRA and XLNET, followed with a post processing method can reach the best ndcg@10, while an ensemble of only BERT large models performs best on mAP.

In this paper, we describe our methods based on pre-trained models for passage re-ranking subtask of Trec 2020 Deep learning track. We find that there is no difference between different kind of models. Also, an ensemble of different pre-trained models can reach higher score than each single model. However, we can not achieve big improvement than the best team in last year.

References

[1] Devlin Jacob, Chang Ming-Wei, Lee Kenton, and Toutanova Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[2] Wang Wei, Bi Bin, Yan Ming, Wu Chen, Bao Zuyi, Xia Jiangnan, Peng Liwei, and Si Luo. Structbert: Incorporating language structures into pre-training for deep language understanding. In *Trec*, 2019.

[3] Yan Ming, Li Chenliang, Wu Chen, Bi Bin, Wang Wei, Xia Jiangnan, and Si Luo. Idst at trec 2019 deep learning track: Deep cascade ranking with generation-based document expansion and pre-trained language model. In *Trec*, 2019.

[4] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. 2019.

[5] Lan Zhenzhong, Chen Mingda, Goodman Sebastian, Gimpel Kevin, Sharma Piyush, and Soricut Radu. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*, 2020.

[6] Clark Kevin, Luong Minh-Thang, Le Quoc V., and D. Manning Christopher. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[7] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[8] Pennington Jeffrey, Socher Richard, and Manning Christopher. Glove: Global vectors for word representation. In *EMNLP*, 2014.