CCNU_IRGroup @ TREC 2019 Deep Learning Track

Hao Hu[1], Junmei Wang[2], Xinhui Tu[1] and Tingting He[1]

[1]School of Computer Science, Central China Normal University, Wuhan, China

[2]School of Mathematics and Statistics, Central China Normal University, Wuhan, China

hu_ha0@qq.com, wjm2018@mails.ccnu.edu.cn, {tuxinhui, tthe}@mail.ccnu.edu.cn

1 INTRODUCTION

The deep learning track consists of two tasks: passage ranking and document ranking. The former focuses on long text retrieval, while the latter focuses on short text retrieval. Both tasks use a large human-labeled set, which is from the MS-MARCO dataset. For different emphases of the two tasks, we adopt two different BERT-based retrieval models. In Section 2 and 3, we will introduce our methods in details. In Section 4 and 5, we will discuss the experiments setting and results.

2 PASSAGE RANKING TASK

Given BERT's excellent performance in a broad range of NLP tasks, we wondered whether we could take the context-dependent token representation learned by BERT to improve the performance of an neural information retrieval model. Recently, many researches have proposed to apply BERT to down-stream tasks through a feature-based method and have shown good performance, such as SDNet [8]. In this task, we only modified the output layer of SDNet to accommodate the retrieval task. Next, we will introduce the output layer of our model. As for the other layers, please refer to SDNet [8].

In the output layer, we first calculate the cosine similarity of a query representation and each passage token representation generated by SDNet based on the features extracted by BERT. Secondly, we select top-k signals and project them into a multi-layer perceptron to get the final decision score [2].

$$u^q = \sum_i \beta_i u_i^q, \qquad \beta_i \propto \exp(w_u u_i^q)$$

$$\mathrm{s} = \mathrm{softmax}((u^q)^T W_s \mathrm{V})$$

$$S_{BERT-feature} = \mathrm{MLP}(\mathrm{topk(s)})$$

where u_i^q denotes the i-th query token representation. w_u and W_s are parameters to be learned. V is a matrix whose column vector is a passage token representation.

BERT's pre-training on surrounding contexts favors text sequence pairs that are closer in their semantic meanings [1,4,6]. Guo et al. [3] discussed the differences between relevance matching and semantic matching. They argue that the ad-hoc retrieval task is mainly about relevance matching, such as exact matching signals, query term importance, and diverse matching requirement. In order to capture relevance matching signals, we combine the BM25 model with our neural IR model. The final relevance score can be calculated as follows:

$$\mathrm{Score} = \alpha \cdot S_{BM25} + (1 - \alpha) \cdot S_{BERT-feature}$$

where S_{BM25} is the original BM25 score and S_{BERT} is the feature-based BERT ranking score. The hyperparameter α can be tuned via cross-validation.

2.1 Model Training

We use a hinge loss to train our model and the loss function is defined as:

$$\mathrm{L}(q, p^-, p^+, \theta) = \max(0, 1 - S_{BERT-feature}(\mathrm{q}, p^+) + S_{BERT-feature}(\mathrm{q}, p^-))$$

where $S_{BERT-feature}(\mathrm{q}, p^+)$ denotes the relevance score for a query and a relevant passage and $S_{BERT-feature}(\mathrm{q}, p^-)$ is the score for a query and a irrelevant passage. θ includes the parameters in this neural model.

3 DOCUMENT RANKING TASK

Yang et al. [7] provide a solution for long document retrieval. Based on the hypothesis that a document is related to a query if some sentences in the document are related, Yang et al. [7] first splits the document into several sentences and calculates the similarity between each sentence and a query, and then selects the top-k scoring sentences.

$$\mathrm{Score}_d = \beta \cdot S_{doc} + (1 - \beta) \cdot \sum_i^n w_i S_i$$

where S_{doc} is the matching score calculated by the traditional retrieval model and S_i is the i-th top sentence score according to BERT fine-tuned on sentence-level dataset. The hyperparameter β and w_i can be tuned via cross-validation. In this task, We reproduce this model and choose BM25 as the traditional retrieval model to calculate S_{doc}.

4 EXPERIMENTAL

4.1 Datasets

4.1.1 Passage ranking datasets.

In order to calculate the BM25 score, we build the index over the entire passage collection file which includes 8.8 million passages. Both the indexing and the BM25 scoring process are accomplished on the Parrot, which is a Python-based Interactive Platform for Information Retrieval [5]. We randomly extract 10% of the data, about 9.7 million passages, from triples.train.small.tsv to build the training set.

4.1.2 Document ranking datasets.

Multi-genre Natural Language Inference (MNLI) is used as the fine-tuning corpus. MNLI is a large-scale, crowdsourced, implicit classification task.

4.2 Settings

We use the BERT-Large [1] in both subtasks. We set the parameters b=0.4, k1=0.9, k2=8 in the BM25 model.

4.2.1 Passage ranking task

We use Adam with learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and use a dropout probability of 0.4 on all layers to train our model, but freeze all parameters of BERT when train the model and we change the maximum length of a sentence in BERT to 256. We select top 10 matching signals in the output layer. The linear interpolation weight $\alpha = 0.8$.

4.2.2 Document ranking task

We use a sliding window of length 100 to split the document into sentences and select the top 3 score sentences to calculate the final score. The linear interpolation weight $\beta = 0.9$.

5 RESULTS

Our methods' performance can be observed in tables 1 and 2.

Table 1: BM25+SDNet+feature-based BERT results in Passage Ranking Task, compared to summary statistics across the 37 submitted runs

Run	MAP	nDCG	P@10
Runid2	0.2781	0.5492	**0.6163**
TREC Median	0.3864	0.6457	0.5651

Table 2: BM25+ fine-tuning BERT results in Document Ranking Task, compared to summary statistics across the 38 submitted runs

Run	MAP	nDCG	P@10
Runid1	0.2366	0.4299	0.5977
TREC Median	0.2989	0.5393	0.6906

We can see that we have not achieved good results on both tasks. After analysis, we argue that there are two reasons for the failure in the passage ranking task. The first one is that we only use 10% of the data to train the model. The second reason is that we only consider the matching of tokens, and ignore the matching of sentences. However, as we know that BERT can capture the matching features between two sentences very well, because BERT is trained to predict whether the next sentence is true or not in the pre-training process. And the reason for failure in the document ranking task is that we do not fine tune the BERT on MSMARCO datasets.

REFERENCE

[1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] Fan Y, Guo J, Lan Y, et al. Modeling diverse relevance patterns in ad-hoc retrieval[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, 2018: 375-384.

[3] Guo J, Fan Y, Ai Q, et al. A deep relevance matching model for ad-hoc retrieval[C]//Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, 2016: 55-64.

[4] Qiao Y, Xiong C, Liu Z, et al. Understanding the Behaviors of BERT in Ranking[J]. arXiv preprint arXiv:1904.07531, 2019.

[5] Tu X, Huang J, Luo J, et al. Parrot: A Python-based Interactive Platform for Information Retrieval Research[C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2019: 1289-1292.

[6] Tenney I, Das D, Pavlick E. Bert rediscovers the classical nlp pipeline[J]. arXiv preprint arXiv:1905.05950, 2019.

[7] Yang W, Zhang H, Lin J. Simple applications of bert for ad hoc document retrieval[J]. arXiv preprint arXiv:1903.10972, 2019.

[8] Zhu C, Zeng M, Huang X. Sdnet: Contextualized attention-based deep network for conversational question answering[J]. arXiv preprint arXiv:1812.03593, 2018.