Higway BERT for Passage Ranking

Di Zhao
University of Delaware
zhaodi@udel.edu

Hui Fang
University of Delaware
hfang@udel.edu

October 28, 2019

1 Introduction

This is the first year of TREC for deep learning and there are two tasks which are document ranking and passage ranking. Our task is passage ranking which can be described as given a query q and the 1000 most relevant passages $p1, p2, ..., p1000$ retrieved by BM25, passage ranking task expects to come up with a successful system to rank the most relevant passage as high as possible.

In this work, we build a neural network based IR model for the passage ranking tasks. Basically we are trying to combine BERT [2] and highway network [6] to get a good performance on the task. Since BERT is the current SOTA model on several nlp related tasks and highway network [6] is kind of neural network with gate structure, since gate structure has already shown its success on some neural network models like lstm [3] and gru [1] on sequence modeling tasks. So we assume combining multi-head attention based transformer with gate based network structure to improve the model performance can be achieved. Meanwhile we also try different loss functions and different training strategies also with axiomatic thinking [5] approach.

2 Problem Setup and Data

TREC passage ranking task uses the data from msmarco passage re-ranking task which provides multiple data format for an easy training on deep learning models. The training data file we use is triples.train.small.tsv, the development data file is top1000.dev.tsv, the evaluation data file is top1000.eval.tsv. The training data file contains three components in each line which are query text, relevant document text and irrelevant document text. The development data contains four component in each line which are query id, document id, query text and document text. The same format goes with top1000.eval.tsv evaluation data file. The development file and evaluation file are retrieved by BM25, there are around 1000 documents retrieved for each query, so basically the task can be a re-ranking task based on BM25 or a full ranking task if the participant can develop their own first step model. The correctly matched query document

pairs are stored in the file called qrels.dev.tsv. So after the model ranking the retrieved documents for each query, it will find the position of the correct document and calculate the score based on the mean reciprocal rank(MRR) metric. The higher the MRR score, the better the model performance.

3 Our Methodology

The model we come up with in this task is called Highway BERT, which is composed by a BERT and a highway network. Since BERT has shown its advantages on several text related tasks such as machine translation and language understanding. We use BERT as a feature extractor to extract the sentence embedding from the query-document pair [4], then use a highway network as a classifier based on the features extracted by BERT and distinguish if the sentence embedding comes from the relevant document or irrelevant document [4]. The reason we choose highway network is because it is a gated network, the gate structure has shown its advantages on some rnn models like lstm and gru. The gate structure can filter the redundant information and only leave the most important information for classification, so basically we want to combine the advantages of the old and new neural network based sequential models to boost the model performance. The highway BERT will classify the query-relevant document as a positive class and classify the query-irrelevant document as a negative class, we use a softmax layer as the last layer which tries to distinguish the two classes as clear as possible. The model structure is shown in Figure 1, in which the left model is highway BERT model with a softmax layer and cross entropy loss, the right figure is the highway BERT model with a ranking loss, the ranking loss ranks the score based on the value of the first dimension of the output sentence embedding. The BERT model comes from the google pre-trained BERT which has twelve encoders, the encoder has the same structure with the transformer encoder, in which there are three components: self-attention, feed forward neural network and residual structure as shown in Figure 1. I add the highway network on top of the pre-trained BERT model and fine-tune it with TREC training data set, the loss function chosen is marginal ranking loss function and cross entropy loss function. The marginal ranking loss function will maximize the distance between the sentence embedding vector of query-relevant document and the query-irrelevant document which means to distinguish them as much as possible, the highway BERT model with ranking loss is represented as BERTH-R. Meanwhile we also use a cross entropy loss on the highway BERT model which is represented as BERTH-C, the BERTH-C model formulates the re-ranking problem to a classification problem.

Besides the change on the model structure, in order to have a better performance we are trying to enrich the model with some human knowledge, we choose the axiom approach [5] in information retrieval to make some perturbation data set for training. Basically we make the perturbation data set based on rule TFC1 which says that a document with more words in query should be more relevant to the query, so we sample a word from the query and add it to the

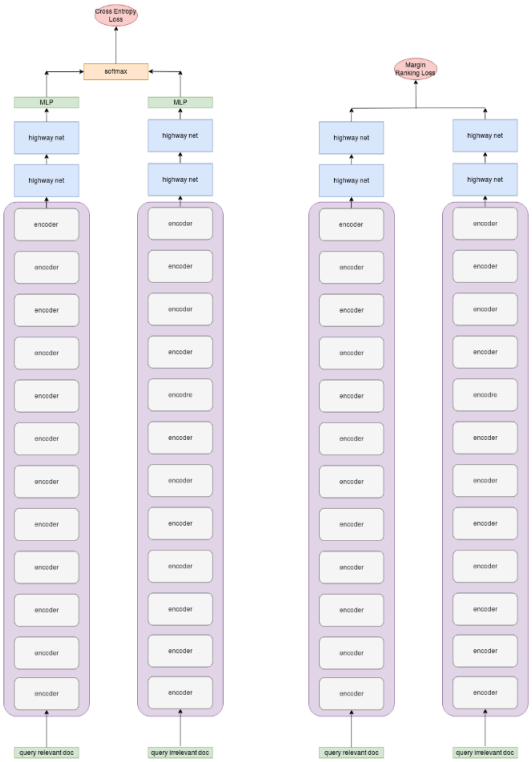

Figure 1: Highway BERT Model Structure.

document as a positive perturbed document Dp, then we add a word that does not exist in the query to the same document and take it as a negative perturbed document Dn. Then like the same procedure described above, we combine the query and document to a query-document pair and feed it into the highway BERT model, since the experiment shows that the performance of the BERTH-C is better than BERTH-R, so we are trying to use the BERTH-C model as a base model for the training on perturbation data set. After training the model with cross entropy loss on the training data set, we use the perturbation data to train the BERT model again with the axiomed ranking loss function [**?**]. The loss function is made up of three marginal ranking loss function. The first one to maximize the margins between the relevant document and the irrelevant document, the second marginal loss function is trying to maximize the margins between the perturbed positive document with the original positive document, and the third marginal loss function is maximizing the margin between the perturbed negative document and negative document. The difference between the three marginal loss is at their margins since the margin for the first marginal loss is bigger which means the model focus more on the original data set rather

than the perturbed data set.

$$
\begin{aligned}
\ell_{AR}(q, d_{pos}, d_{neg}, \Delta_i) = & \\
& \max\{0, \epsilon - (s_\theta(q, d_{pos}) - s_\theta(q, d_{neg}))\} \\
& + \lambda \cdot \max\{0, \mu - \delta_i \cdot (s_\theta(q, d_{pos}) - s_\theta(q, \mathrm{d}^{(i)}_{pos}))\} \\
& + \lambda \cdot \max\{0, \mu - \delta_i \cdot (s_\theta(q, d_{neg}) - s_\theta(q, \mathrm{d}^{(i)}_{neg}))\}
\end{aligned}
$$

Figure 2: Axiom Equation.

4 Experiment

There are three models running independently in the experiment which are 1.Highway BERT with ranking loss function, 2.Highway BERT with cross entropy loss function, 3.Highway BERT with cross entropy loss function trained on perturbation data set. The working pipeline is referred from the CKNRM of thunlp group [1]. The model 1, 2 are running on a Tesla P100 with training batch size as 10 and 6000 training steps before validation. It takes about 1 hour to train 60000 data samples and about 20 hours to rank the 7 million documents in the top1000.dev.tsv file, the learning rate is set to 3e-3. The model 3 is running on google cloud platform with a Tesla V100 gpu, the training batch size is 10 and 6000 training steps before validation. It takes about 1 hour to train 60000 data samples and 12 hours to rank the 7 million documents in the top1000.dev.tsv file, the learning rate is set to 3e-5.

For the axiomed highway BERT, I tried several different training strategies. The first training strategy is training the pre-trained google BERT model directly on the perturbation data set with the axiomed loss function. The second training strategy is training the pre-trained google BERT model with the original msmarco data and the perturbed documents and queries with a single ranking loss. The third training strategy is the same as the description above which uses msmarco data with cross entropy loss function to train the google pre-trained BERT model after it reaches MRR at 0.347, then I change the msmarco data to the perturbation data with the axiomed ranking loss function. Basically, the model performance of the first and second training strategy are very low, for the third training strategy finally I get MRR at 0.347 which means there is no performance improvement. However I assume that even the training strategy cannot improve the model performance, it can diversify the model which means the axiomed highway BERT may rank the correct document at different position comparing with the highway BERT model without training on perturbation data set, this diversification may work on the ensembles of highway BERT model. As the Table 1 shows that the ensemble highway

[1] https://github.com/thunlp/Kernel-Based-Neural-Ranking-Models

BERT model (BERTH-E) does get a performance improvement, it seems that the highway BERT trained with axiomed data does give a diversified ranking on the documents.

The results of the models are shown below:

	BERTH-A	BERTH-R	BERTH-C	BERTH-E
MRR@10	0.336	0.326	0.336	0.339

Table 1: Experiment Result Table.

5 Conclusion

From this task we find that BERT model has a good performance on passage ranking task, besides we realize that the pre-train and fine-tune two step process is a powerful training strategy. So here are some open questions: Does more parameters mean better performance for deep learning? What is the good way to simplify BERT to make it run faster and better? And how to apply IR axiomatic thinking approach to leverage the performance of these big models which means can we find the preference or weakness of these models?

References

[1] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS*, 2014.

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

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 1997.

[4] Yifan Qiao, Chenyan Xiong, Zheng-Hao Liu, and Zhiyuan Liu. Understanding the behaviors of BERT in ranking. *ArXiv*, 2019.

[5] Corby Rosset, Bhaskar Mitra, Chenyan Xiong, Nick Craswell, Xia Song, and Saurabh Tiwary. An axiomatic approach to regularizing neural ranking models. *SIGIR*, 2019.

[6] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *ICML*, 2015.