CS475 Fall 2022 Homework 3

Team 4: Hong Seok Kang, Kyung Wook Nam, Hyoung Jo Bhang

October 12, 2022

1 Introduction

`BERT` [2] based models for sentence-level tasks mainly use feature vector generated from special token `[CLS]`. However, various studies were conducted to utilize information behind the hidden states of models, by suggesting different pooling methods.

In [5], `[CLS]-`, `[SEP]-`, `Mean-`, `Max`-pooling were applied to `SentEval` on BERT, and `Mean`-pooling performed the best. A follow-up study [4] proposed a pooling layer which combines `[CLS]` token and `MeanMax`-pooling. It significantly improved F1 score on Large-Scale Multi-label Text Classification task. Moving further, a computer vision paper [3] suggested `K-MAX Pool`, which computes mean of the top-K maximum values among elements, and [1] implemented Generalized Pooling Operator, which learns the best pooling strategy with positional encoding.

In this report, we compared `class BertPooler`, which only takes [CLS] token, to `class MeanMaxTokensBertPooler` and our `class MyBertPooler` on `MRPC` dataset of `GLUE` [6]. Our approaches on pooling are as follows: First, the impact of each pooling method was analyzed in detail. Then, we applied `K-MAX Pool` method [3] to our task, previously not widely used in NLP. In result, `class MyBertPooler` showed better result than `class MeanMaxTokensBertPooler`, and with brief analysis on the table, the performance of the methods are discussed with some possible future approaches.

2 Data

For experiment, we used the `Microsoft Research Paraphrase Corpus (MRPC)`, which evaluates semantic textual similarity between sentences. Each sentence pair is labeled *equivalent* or *not_equivalent* by human annotators. The whole set is divided into a training subset (4,076 sentence pairs of which 2,753 are paraphrases) and a test subset (1,725 pairs of which 1,147 are paraphrases).

3 Methods

In this section, we explain three main metrics that are used in a series of experiments we conducted on the way to build our own pooler (`class MyBertPooler`).

CLS (classification) is an WordPiece Embedding vector located at the start of each sentence. It is a special token in BERT to train for 'Next Sentence Prediction', introduced to solve important tasks in NLP, such as QA (Question Answering) or NLI (Natural Language Inference).

However, BERT is designed in such a way that each vector reflects information from all other word vectors. The difference is `[CLS]` does not hold any additional information of a word, but only positional data. Due to its clarity and simplicity, numerous papers using BERT adopt it.

MeanMax Details of `MeanMax` could be found in the assignment description. Implemented as a simple linear concatenation of `Mean` and `Max`, our team has separated each method to understand the effectiveness of each method, respectively.

TopKMean `Max`-pooling method only focuses on the *one best* feature, extracting and maximizing the impact of the unique feature from surroundings. It has been a popular method applied in various vision tasks, but this is not an ideal method when it comes to paraphrasing. In paraphrasing tasks, the goal is to

analyze whether important features on one sentence is retained in another without omissions or modifications. Therefore, `TopKMean` was selected for our approach, calculating the mean of top K features, while disregarding other unnecessary features.

$$C_{\mathrm{MMT}} = \left[\sum_{i=0}^{K-1} topK_i(T_i)\right]/K, \quad C_{\mathrm{MMT}} \in \mathbb{R}^H, \quad T_i \in \mathbb{R}^H, \tag{1}$$

4 Results & Discussions

With the pooling layers discussed earlier, we conducted experiments to test efficiency and coherence of each method. The results in `Table 1` are averaged values after 10 times of evaluations.

Pooling layers	Loss	Accuracy	F1 score[%]
`[CLS]`	0.3902	0.8510	90.18
`MeanMax`	0.3495	0.8525	89.31
`Mean`	0.3468	0.8598	89.89
`Max`	0.3672	0.8417	88.48
`TopKMean (10)`	0.3784	0.8423	88.56
`TopKMean (20)`	0.3432	0.8627	90.02
`TopKMean (30)`	0.3654	0.8593	89.86
`TopKMean (half)`	0.3700	0.8534	89.52

Table 1: Performance of each pooling layers on MRPC data set.

4.1 CLS, MeanMax

`[CLS]` shows the highest F1 score. Scores of `Mean` and `Max` separated prove our hypothesis that `Max` does not capture all the necessary information in a sentence. `Max` exhibits the lowest F1 score among all metrics and we conclude that it is not appropriate for paraphrasing tasks.

4.2 TopKMean

Experiments on this method were conducted with the K value of 10, 20, 30 and half of sequence length (L). The highest F1 score was obtained when $K = 20$. `TopKMean` pooling methods also showed better performance than `Mean` and `Max` layers. `TopKMean` is an improved version of `Max`-pooling, capturing top k features (words) that characterize the sentence.

In our experiment, due to limited time and resources, only several values of K were tested for comparison, carefully arranged according to our dataset. It exhibits the potential of `TopKMean` pooling methods in paraphrasing tasks, adjusting its value in line with data and sentences used.

Therefore, in future research, it will be very interesting to analyze the result from different K values along with the changes of vector representation in accordance with sentence and data. Focusing on characteristics such as: length, structure, type (academic, novel, ..), topic (political, science, environment, ..). Additionally, investigating the relation between K and $L - K$ removed features ('What can we ignore?') could be another approach. Extending from [1], implementation of Generalized Pooling Operator also could be another task, which learns the best pooling strategy by reflecting positional information of tokens in sentences.

Last but not least, ensembles of pooling methods is a large area to look over. In our experiment, `MeanMax` showed a score that lies in between the performance of two methods run individually. It could seem that the score lies in the expected, reasonable scope. However, other evaluations of ensembles, not stated in this report, did not present clear tendency in results. We believe further study on ensembles is necessary, searching for the combination metric that harmonically aligns the advantages of different pooling methods.

References

[1] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, page 15789–15798, 2021.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[4] Jan Lehečka, Jan Švec, Pavel Ircing, and Luboš Šmídl. Adjusting bert's pooling layer for large-scale multi-label text classification. In *International Conference on Text, Speech, and Dialogue*, page 214–221. Springer, 2020.

[5] Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. Universal text representation from bert: an empirical study. *arXiv preprint arXiv:1910.07973*, 2019.

[6] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.