Exploring Strategies for Improved Performance in Multi-Task Learning with Pretrained-BERT

Stanford CS224N Default Project

Xiaolei Shi
Stanford Center for Professional Development
Stanford University
xiaoleis@stanford.edu

Abstract

We proposes a multi-task learning approach utilizing pretrained-BERT for Sentiment Analysis, Paraphrase Detection, and Semantic Textual Similarity. The approach involves a three-step process: BERT Further Pre-training, Classifier Pre-training, and Multi-Task Fine-tuning. To improve performance on the target tasks, two strategies were implemented: advanced model structures and further pretraining BERT with task-specific datasets. Experimental results demonstrate that pretraining BERT with in-domain data significantly enhances performance on target tasks while maintaining performance on other tasks. Moreover, the use of a complex classification head did not enhance target task performance without sufficient training data.

1 Key Information to include

- Mentor: Sauren Khosla
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Pretrained models, such as BERT Devlin et al. (2018) and GPT Radford et al. (2018), have proven to be highly beneficial for a range of Natural Language Processing (NLP) tasks. Rather than requiring a separate model for each specific task, a single model that performs equally well on multiple downstream NLP tasks is highly preferred. The primary advantage of this multi-tasking approach is that these tasks utilize the same model to process input sentences and obtain sentence encodings, with the only difference being the customized head required for each task to generate output in the desired format, such as a class label for text classification or a logit for textual similarity detection. Consequently, sharing a pretrained model, such as BERT, as the encoder and attaching a header for each task is a common approach.

In this work, we have adopted this philosophy to build a multi-task model based on pretrained-BERT to handle three tasks: Sentiment Analysis (SA), Paraphrase Detection (PD), and Semantic Textual Similarity (STS). Our experimental results indicate that this model performs well on PD and STS, but is only marginally satisfactory on SA. To improve performance on these target tasks, we explored two strategies, including advanced model structures and further pretraining BERT using task-specific datasets. Our experiments indicate that further pretraining BERT with in-domain data can significantly improve performance on the target task, without harming the performance on other tasks. Furthermore, we found that using a complex classification head did not enhance performance on the target tasks.

3 Related Work

Multi-task learning (MTL) (Caruana (1993)) has been widely used for models designed to handle multiple tasks. Many existing works (Sun et al. (2019), Liu et al. (2019a)) have utilized BERT as a shared text encoding layer in a multi-task model. While our approach trains all tasks together but with independent loss functions for each task, we acknowledge the value of other MTL approaches in improving model performance.

In addition to MTL, various pretraining strategies for BERT have been explored to enhance downstream task performance (Liu et al. (2019b)). For instance, larger batch sizes over more data, removing the next sentence prediction objective, training on longer sequences, and dynamically changing the masking pattern applied to the training data have been shown to be effective pretraining techniques.

Other works have demonstrated the significant benefits of within-task and in-domain pretraining for further improving the performance of downstream tasks (Sun et al. (2019)). Additionally, a more complex autoregressive pretraining method that enables learning of bidirectional contexts and integrates ideas from Transformer-XL (Liu et al. (2019a)) into pretraining has outperformed BERT by a substantial margin on a wide range of NLP tasks.

Furthermore, some works have investigated transfer learning approaches beyond BERT, such as XLNet (Yang et al. (2019)), ELECTRA (Clark et al. (2020)), which achieve state-of-the-art performance on various NLP tasks. These models have improved upon BERT by modifying the pretraining objective, introducing additional pretraining tasks, or using larger amounts of training data.

4 Approach

4.1 Architecture

The BERT-base model (Devlin et al. (2018)) comprises an encoder with 12 Transformer blocks, each with 12 self-attention heads, and a hidden size of 768. The maximum input sequence length is 512 tokens. The input sequence may have one or two segments, and the first token is always [CLS], which contains a special classification embedding. The [SEP] token is used to separate segments. BERT uses the final hidden state h of the [CLS] token as the representation of the whole sequence.

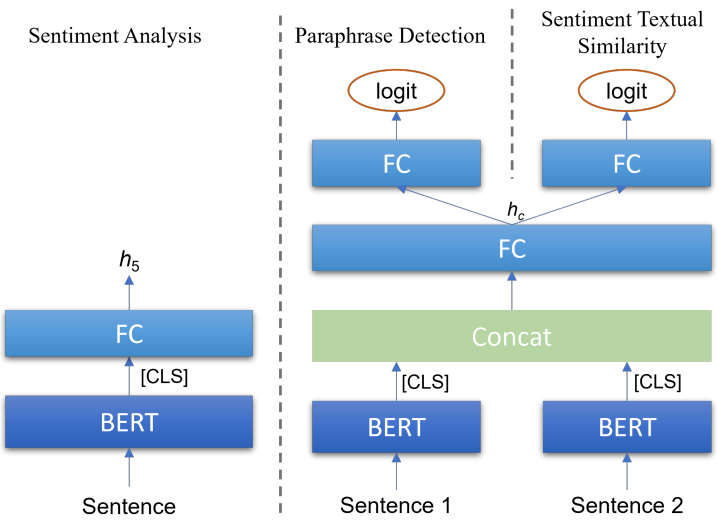

Figure 1: Model architecture of the multi-task NLP classifier based on BERT

As shown in Figure 1, the BERT encoder is augmented with three task-specific heads to perform the downstream tasks. For the sentiment analysis (SA) task, a softmax classifier is used to predict the probability c of a label from a set of five possible labels (negative, somewhat negative, neutral, somewhat positive, or positive). The classification is performed on the output of the [CLS] token:

$$p(c|h) = \texttt{softmax}(W_{\texttt{sa}} h_{\texttt{cls}})$$

where W_{sa} is the task-specific parameter matrix.

For the paraphrase detection (PD) and semantic textual similarity (STS) tasks, two sentences are separately fed into BERT to generate their respective [CLS] tokens. The two [CLS] tokens are concatenated and fed into a shared fully connected layer to produce the hidden state h_c. The PD task uses a binary classifier on h_c to determine whether two phrases convey the same semantic meaning:

$$c = \texttt{logistic}(W_{\texttt{pd}} h_c)$$

where $W_{\texttt{pd}}$ is the parameter matrix of the output layer for the PD task.

The STS task outputs a scale from 0 (not at all related) to 5 (same meaning) that measures the degree of semantic equivalence between two sentences:

$$r = W_{\texttt{sts}} h_c$$

where $W_{\texttt{sts}}$ is the parameter matrix of the output layer for the STS task. Note that the parameters of the shared concatenation and MLP layer are also shared between the PD and STS tasks.

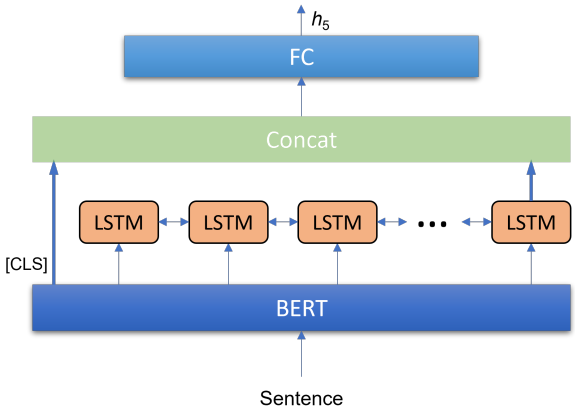

Figure 2: LSTM-based classifier for the Sentiment Analysis task

In order to examine the impact of model structure on performance, we will employ a bidirectional LSTM base model as a competitor to the simple softmax classifier for the SA task. The model architecture is depicted in Figure 2. Specifically, the tokens in the final output layer, excluding the [CLS] token, are fed into a bidirectional LSTM layer. The resulting final output of the LSTM layer is concatenated with the [CLS] token, and subsequently passed through a fully connected layer. Finally, an output softmax layer is utilized to generate predictions for each label.

4.2 Methodology

When applying BERT to downstream NLP tasks, it is crucial to carefully determine the finetune strategy. In this study, we adopt a three-step approach for a delivered model: 1) Further Pre-training, 2) Classifier Pre-training, and 3) Multi-Task Fine-tuning, the training flow is shown in Figure 3.

Firstly, we further pretrain BERT using the target task-specific dataset, which exhibits a different data distribution compared to the dataset adopted by BERT for pretraining. To evaluate the effectiveness of this step, we perform a parallel experiment without this step, and the resulting performance serves as the baseline for comparison.

Secondly, we freeze the parameters of BERT and only train the classification heads of each task. The updated weights serve as a sound initialization for further finetuning. We also investigate the impact of model structure by replacing the trivial classifier with an LSTM-based model, as described in section 4.1.

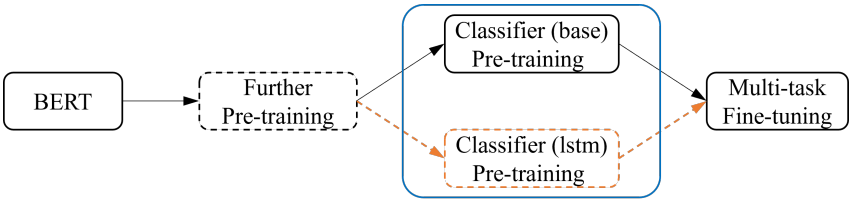

Figure 3: Training flow of the multi-task NLP classifier based on BERT

Finally, we conduct Multi-task Fine-tuning by training all tasks together to update the weights of both BERT and the classification heads, while each head has an independent loss function. Our expectation is that BERT will leverage the shared knowledge among tasks.

5 Experiments

In this paper, we explore the impact of two factors on the performance of a multi-task classifier for Sentiment Analysis (SA), Paraphrase Detection (PD), and Sentiment Textual Similarity (STS) tasks. Firstly, we investigate the effect of further pretraining the BERT model with task-specific data. Secondly, we examine how a more complex model structure influences the performance of the multi-task classifier. We utilize the official pretrained uncased BERT-base model as the starting point of our training flow, as illustrated in Figure 3.

5.1 Datasets

For each of the three classification tasks, we train our models on widely-studied datasets. However, we note that these datasets have varying sizes. In particular, the Quora dataset[1] for the PD task is significantly larger than those of the other two tasks. As a result, the trained model may exhibit bias towards the PD task, as demonstrated in Section 5.3. We present the statistics for each dataset in Table 1.

Dataset	# of Train	# of Dev	# of Test
SST	8,544	1,101	2,210
Quora	141,506	20,215	40,431
STS	6,041	864	1,726

Table 1: Summarization of dataset splits

Sentiment Analysis We use the Stanford Sentiment Treebank (Socher et al. (2013)), which contains 215,154 unique phrases extracted from movie reviews. Each phrase is annotated by three human judges with one of five sentiment labels: negative, somewhat negative, neutral, somewhat positive, or positive. Given the discrete labels of the this dataset, we adopt accuracy as the metric to test accuracy on this dataset.

Paraphrase Detection We use the Quora dataset released by the Quora website, which consists of 400,000 question pairs labeled as paraphrases or not. Given the binary labels of this dataset, the metric that we utilize to test this dataset is accuracy.

Sentiment Textual Similarity We use the SemEval STS Benchmark Dataset (Agirre et al. (2013)), which contains 8,628 sentence pairs with varying degrees of similarity on a scale from 0 (unrelated) to 5 (equivalent meaning). When testing this dataset, we calculate the Person correlation of the true similarity values against the predicted similarity values across the test dataset.

Further Pretraining BERT We employed the Large Movie Review Dataset (Maas et al. (2011)) to further pretrain BERT. To construct the training dataset, we first split each independent review into sentences and processed them into samples, with each sample consisting of two segments, or sequences of tokens. To ensure consistency in the training dataset, we set a maximum sequence length of 512 for each sample and truncated longer reviews accordingly. To facilitate the masked

[1] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

language modeling (MLM) task, we randomly masked 15% of tokens in each sequence during the construction of the training dataset. Specifically, for a given token to be masked, we replaced it with the special [MASK] token 80% of the time. In 10% of cases, we retained the original token, and in the remaining 10% of cases, we replaced it with a randomly chosen token. For the next sentence prediction (NSP) task, we constructed each sample by taking two segments, with 50% of the samples being from continuous sentences and the other 50% from different reviews. The following shows a short sample from our constructed dataset. The evaluation and loss calculation are the same with (Devlin et al. (2018)).

> [CLS] following the es ##cap ##ades of kei [MASK] [MASK] la ##tri ##na and nate ##lla , our three " [MASK] " for [MASK] of a better term , the show [MASK] ' t shy away [MASK] parody ##ing [MASK] im ##agi ##nable subject [MASK] political correct ##ness flies out the window in ##under episode. [SEP] i didn ' t know this came from canada , but it is very good . very good ! [SEP]

5.2 Hyperparameters

We employ the BERT-base model (Devlin et al. (2018)), featuring a hidden size of 768, 12 Transformer blocks (Vaswani et al. (2017)), and 12 self-attention heads. During further pretraining with BERT, we use 1 NVIDIA V100 GPU, set the batch size to 16, max sequence length to 512, and learning rate to 2e-5, while training for 5 epochs.

For pretraining the three classification heads, we train on an NVIDIA H100 GPU for 48 epochs, with a fixed learning rate of 1e-3 and batch size of 2048 to fully utilize the GPU memory, while keeping the BERT parameters frozen.

For the multi-task fine-tuning, we train on an H100 GPU for 48 epochs, with a learning rate of 1e-5 and batch size of 160. Throughout all training stages, we set the dropout probability to 0.1, and employ the Adam optimizer (Kingma and Ba (2015), Loshchilov and Hutter (2017)) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a decay rate of 0.01.

5.3 Exp-I: Investigate the Further Pretraining

In this section, we examine the effectiveness of further pre-training BERT on the movie review dataset by leveraging unsupervised masked language modeling and next sentence prediction tasks. The Stanford Sentiment Treebank and the movie review dataset share the same distribution, being derived from movie reviews. Thus, our primary focus is on the difference in accuracy for the Sentiment Analysis task.

	Acc. of SA Task	Acc. of PD Task	Person Correlation of STS Task
w/o Further Pretrain	0.480	0.851	0.623
W/ Further Pretrain	0.549 (14.4%↑)	0.858 (0.8%↑)	0.631 (1.3%↑)

Table 2: Test set accuracy for downstream tasks of w/o and w/ further pretraing BERT using task-specific dataset

The results are presented in Table 2, where we observe a substantial improvement in the Sentiment Analysis task as compared to the original BERT-base model (row 'w/o further pretrain' in Table 2). The accuracy increased by 14.4%, while the other two tasks show only marginal improvements (1%). Given that BERT is pretrained on the general domain, which has a distinct distribution from the movie reviewing domain, we argue that further pretraining BERT with the movie review dataset helps BERT acquire knowledge specific to this domain. As a result, BERT exhibits superior performance in classifying the sentiment of movie review phrases. Furthermore, the slight improvement in accuracy for the other two tasks indicates that further pretraining BERT with task-specific datasets does not adversely affect downstream tasks that rely on datasets from other domains.

5.4 Exp-II: Investigate the Model Structure

In this section, we investigate the impact of model structure on the performance of the Sentiment Analysis task by replacing the simple softmax classifier with a more complex LSTM-based classifier with an increased number of parameters. We employ the best strategy from Exp-I during the training process.

Classifer of SA Task	Acc. of SA Task	Acc. of PD Task	Person Correlation of STS Task
Simple	0.549	0.858	0.631
LSTM-based	0.500 (8.93%↓)	0.859 (0.1%↑)	0.656 (4.0%↑)

Table 3: Comparison of test accuracy for downstream tasks between simple and LSTM-based classifier

The findings, presented in Table 3, reveal a 8.93% decrease in performance for the Sentiment Analysis task when compared to the simple classification head. However, the accuracy of the other two tasks slightly increased, despite being trained simultaneously. We hypothesize that the simple classifier based on the [CLS] token is sufficient for the Sentiment Analysis task, and the suboptimal performance is attributable to a lack of training data. This conclusion holds for the Sentiment Textual Similarity task as well. Although the LSTM-based classifier demonstrates strong ability, it cannot accurately classify sentiment labels when provided with inaccurate hidden states as inputs. Furthermore, the addition of more parameters increases the risk of overfitting with insufficient training data.

6 Conclusion

In conclusion, our study investigates two factors that can enhance the performance of BERT in a multi-task classifier. Firstly, we evaluate the effectiveness of further pretraining BERT using task-specific datasets. We find that this strategy significantly improves the downstream performance of BERT, while having no adverse effects on tasks that rely on datasets from other domains. Secondly, we examine the impact of model structure on performance and observe that implementing a classifier with a more complex structure and an increased number of parameters does not enhance accuracy when training datasets are limited. In fact, such a strategy can result in the deterioration of performance due to the heightened risk of overfitting.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *International Conference on Machine Learning*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. In *Proceedings of the 2018 Association for Computational Linguistics (ACL) Conference*, pages 172–186. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification?

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.