minBert and Downstream Tasks

Stanford CS224N Default Project

Zhimin Tang
Stanford University
tangzm@stanford.edu

Abstract

In the field of natural language processing, the use of BERT (Bidirectional Encoder Representations from Transformers) has gained significant attention. In this paper, we explore the application of minBert, and focus on three different downstream tasks: Sentiment Analysis, Paraphrase Detection, and Semantic Textual Similarity.We experiment with various factors including pooling strategy, learning rate, dropout rate, and classification objective function. Specifically, we found the use of mean pooling in Semantic Textual Similarity (STS) get better performance. Besides,by implementing a Siamese-inspired network architecture , we also compare different concatenation technologies.

1 Key Information to include

- Mentor:
- External Collaborators (if you have any):No
- Sharing project:No

2 Introduction

BERT set new state-of-the-art performance on various sentence classification and sentence-pair regression tasks. In the past, research mostly focused on developing individual models that focuses on specific language tasks from scratch, and little knowledge sharing occurs across different models and different tasks. However, large attention-based language models that are heavily trained on simple tasks over a huge corpus of text have proven to produce very powerful token embeddings that significantly benefit almost every major downstream language task. As a result, researchers began to utilize these pretrained models as a starting point to build state-of-the-art models that tackle different downstream language tasks.

This paper aims to explore ways to adapt BERT to three different tasks - sentiment analysis of a sentence, paraphrase detection between sentence pairs, and similarity recognition between sentence pairs.My approach involves refining BERT through a combination of modern techniques, including the implementation of a Siamese-inspired network architecture.

3 Related Work

3.1 BERT Model and Multitasking

Introduced by Jacob Devlin and Kristina (2018), the BERT model has led to state-of-the-art results in many NLP tasks and has significantly reduced the need for labeled data by pretraining on unlabeled data over different pre-training tasks. BERT is first trained trained on plain text for masked word prediction and next sentence prediction tasks. We call this first step pretraining. Then, it is finetuned on a specific linguistic task with additional taskspecific layers using task-specific training data.

3.2 Siamese Network Architectures

Siamese network architectures have been widely used in tasks involving similarity assessment, such as paraphrase detection and sentence similarity recognition. Nils Reimers (2019) introduced SBERT , which produces sentence embeddings that can detect semantic similarity between pairs of sentences .

4 Approach

In this section we will describe the approach we followed.

4.1 Model Architecture

Our model architecture is based on a minimal BERT Devlin et al. (2018) and ADAMW optimizer that we implemented by completing the provided skeleton code. The mean pooling representation of the last_hidden_state token from the last BERT layer output is then used as a sentence representation for downstream classification tasks.

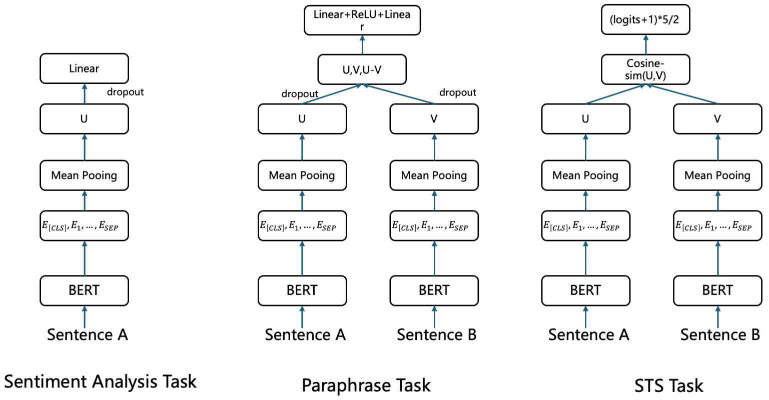


Figure 1: Model architecture for sentiment, paraphrase, and similarity tasks.

4.2 Hypeparameter tuning

We use different dropout rates,learning rates in three task.

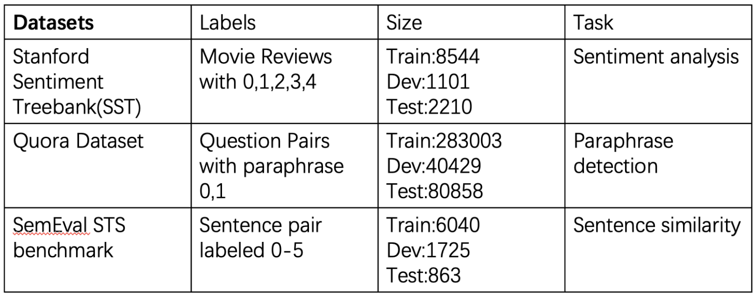
4.3 Classification objective function

For paraphrase task, we experiment different classification objective function and concatenate the sentence embeddings u and v with the element-wise difference |u v| and multiply it with the trainable weight W_t.

5 Experiments

5.1 Data

We are using the provided datasets for the default projects with the following splits :



Datasets	Labels	Size	Task
Stanford Sentiment Treebank(SST)	Movie Reviews with 0,1,2,3,4	Train:8544 Dev:1101 Test:2210	Sentiment analysis
Quora Dataset	Question Pairs with paraphrase 0,1	Train:283003 Dev:40429 Test:80858	Paraphrase detection
SemEval STS benchmark	Sentence pair labeled 0-5	Train:6040 Dev:1725 Test:863	Sentence similarity

5.2 Evaluation method

Since sentiment analysis and paraphrase detection are classification tasks, we use a simple accuracy metric,calculated by dividing percent correctly classified examples by total examples.

As for semantic textual similarity, we use Pearson correlation of the true similarity values against the predicted similarity values for the SemEval STS Benchmark Dataset

5.3 Experimental details

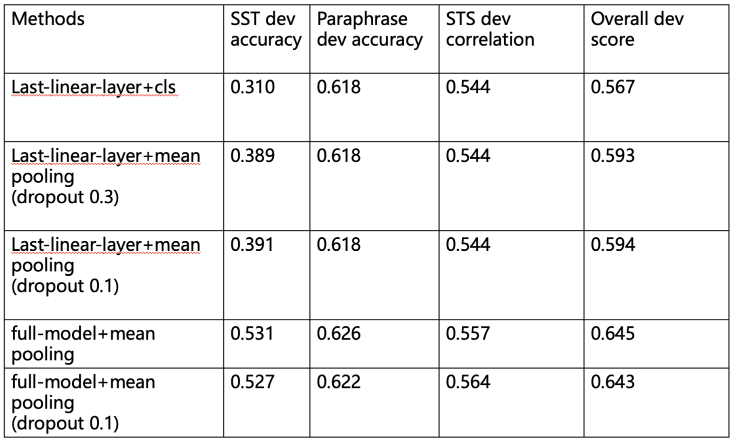
Report how you ran your experiments (e.g., model configurations, learning rate, training time, etc.)

We use a cloud machine NVIDIA P100 with 4vcpu and 30GB mem. The batch size is 32 with P100.The hidden dimension was always 768.

We only train on sst datasets.

5.4 Results

Report the quantitative results that you have found. Use a table or plot to compare results and compare against baselines. We got best dev score 0.645 and test score 0.637



Methods	SST dev accuracy	Paraphrase dev accuracy	STS dev correlation	Overall dev score
Last-linear-layer+cls	0.310	0.618	0.544	0.567
Last-linear-layer+mean pooling (dropout 0.3)	0.389	0.618	0.544	0.593
Last-linear-layer+mean pooling (dropout 0.1)	0.391	0.618	0.544	0.594
full-model+mean pooling	0.531	0.626	0.557	0.645
full-model+mean pooling (dropout 0.1)	0.527	0.622	0.564	0.643

Table 1: Results of multi-task learning on SST, Quora, and STS datasets, last-linear-layer means only task-specific parameters are tuned and full-model means all parameters are tuned.

We observed a 25 percents improvement in SST performance when mean pooling was applied in sentiment analysis. However, changing the learning rate from 0.3 to 0.1 resulted in only a slight and insignificant change in the score.

Methods	SST dev accuracy	Paraphrase dev accuracy	STS dev correlation	Overall dev score
Para+(u,v,u-v)	0.389	0.618	0.544	0.593
Para+(u+v)	0.389	0.578	0.544	0.58
Sts+cos	0.389	0.618	0.544	0.593
Sts+mul	0.389	0.618	0.120	0.522

Table 2: Results of multi-task learning on SST, Quora, and STS datasets,cos means cosinesimilarity between the two sentence embeddings. For the paraphrase classification task, we concatenate two sentence embeddings u and v with the absolute value of their element wise difference |u v|, and feed these three vectors into our classifier head.

We observed a 450 percents improvement in STS performance when cosine similarity was applied.

6 Analysis

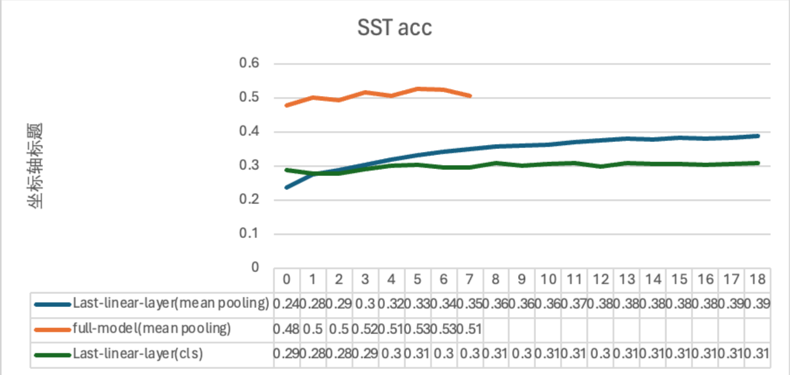
Our findings indicate that Sentence-BERT achieves better performance compared to Base BERT in paraphrase and similarity tasks. There are three key distinctions between Sentence-BERT and Base BERT:

1.Sentence-BERT employs mean pooling of sequence output instead of relying on the [CLS] token for sentence embedding.

2.For the similarity task, Sentence-BERT utilizes cosine similarity instead of a linear classifier.

3.The input for the sequence pair classifier in Sentence-BERT consists of (u, v, |u v|) instead of just (u, v) or (u+v), where u and v represent sentence embeddings.

Besides, we analysis the speed and accuracy between all parameter and task-specific parameter tunning. In general, the common belief is that tuning task-specific parameters while keeping other parameters frozen can lead to faster computations. But as Figure 2 shows, we get best acc in 3 epochs with all parameters tunning. There is a potential risk of overfitting. Including irrelevant or noisy parameters can cause the model to become too complex, leading to a high accuracy on the training data but poor generalization to unseen data.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Last-linear-layer(mean pooling)	0.24	0.28	0.29	0.3	0.32	0.33	0.34	0.35	0.36	0.36	0.36	0.37	0.38	0.38	0.38	0.38	0.38	0.39	0.39
full-model(mean pooling)	0.48	0.5	0.5	0.52	0.51	0.53	0.53	0.51											
Last-linear-layer(cls)	0.29	0.28	0.28	0.29	0.3	0.31	0.3	0.3	0.31	0.3	0.31	0.31	0.3	0.31	0.31	0.31	0.31	0.31	0.31

Figure 2: SST acc with ecochs. it can be observed that the accuracy of SST significantly improves when all parameters are tuned. Additionally, tuning all parameters leads to much faster results compared to solely tuning task-specific parameters.

7 Conclusion

During this project, we employed various techniques to manage multitasking with the aim of identifying key principles and optimal approaches for achieving efficient and rapid training. We gained insights into handling multiple inputs by concatenating sentences, implementing diverse pooling strategies, experimenting with different learning methods, and adjusting batch sizes. Notably, we discovered the effectiveness of mean pooling, cosine similarity in pairs of sentences, and the significance of hyperparameter tuning.

In order to further enhance our project, there are a few future improvements that we plan to implement. We aim to expand our training datasets by incorporating paraphrase datasets and sts datasets. This will help in better understanding the nuances of language and improving the overall performance of the model. Additionally, we intend to balance the proportions between these datasets to ensure a well-rounded and comprehensive training experience.

Furthermore, we plan to fine-tune the loss function used in our training process. By exploring different loss(Matthew Henderson and Kurzweil (2017))functions and their impact on model performance, we can optimize the training process and achieve even better results.

8 Ethics Statement

The ethical challenge of biased sentiment analysis is particularly pertinent to sentiment analysis, as it directly involves interpreting and classifying emotional content in text. If the dataset used to train the model contains erroneous or maliciously labeled information, there is a risk that the model will learn to associate negative, derogatory, or racially charged language with positive sentiment. This misclassification could have severe consequences, such as reinforcing harmful stereotypes or providing skewed analytics that could be used to justify unethical policies or business decisions.

To mitigate this risk, a comprehensive data audit and preprocessing strategy should be implemented. This involves not only cleaning the data for errors but also critically assessing the labels assigned to ensure they do not reflect biased or incorrect sentiments. Employing techniques such as active learning can help identify and correct label inaccuracies by involving human annotators in the loop. Furthermore, it is important to maintain transparency in the dataset curation process and to provide clear documentation so that users of the model can understand the context and limitations of the training data.

In the project centered on paraphrase detection using Quora datasets, one ethical issue arises from the potential for privacy violations. Since Quora is a platform where individuals often discuss personal experiences and viewpoints, the dataset may contain sensitive information that can be traced back to specific users, even if unintentionally. This raises concerns about the right to privacy and the potential misuse of data, where individuals' personal information could be exposed or exploited without their consent.

To mitigate the risk of privacy violations, a rigorous data anonymization process must be implemented. This includes removing or obfuscating any identifiable personal information and ensuring that the dataset cannot be cross-referenced with other databases to re-identify users. Additionally, it is crucial to obtain informed consent from users whose data is being used, making them aware of the scope and purpose of the project, and allowing them to opt out if they wish.

References

Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *arXiv preprint arXiv:1810.04805*.

Brian Strope Yun-Hsuan Sung László Lukács Ruiqi Guo Sanjiv Kumar Balint Miklos Matthew Henderson, Rami Al-Rfou and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. In *arXiv preprint arXiv:1705.00652*.

Iryna Gurevych Nils Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc. that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.