MultiBERT: Enhanced Multi-Task Fine-Tuning on minBERT

Stanford CS224N Default Project

Christina Tsangouri
Department of Computer Science
Stanford University
xristina@stanford.edu

Abstract

This project aims to leverage the capabilities of BERT (Bidirectional Encoder Representations from Transformers), a model pre-trained semi-supervised on a vast corpus of text and further fine-tuned for specific downstream tasks. BERT's core methodology involves a 'masked language model' objective, where it learns to predict randomly masked tokens based on their context. In this project, I complete a implementation of a minimal version of BERT, which is utilized for sentiment analysis on the SST and CFIMD datasets, leveraging the pre-trained weights and subsequently fine-tuning on these datasets. In the project extension I also fine-tune and explore enhancements to the BERT model for use as a multi-task classifier, aiming to achieve improved performance in three NLP tasks (sentiment analysis, paraphrase detection and semantic textual similarity). I experimented with multi-task finetuning, additional single task finetuning post the multi-task training, and using a learning rate scheduler to adjust the learning rate while training. I found good results with: ADD RESULTS

1 Key Information to include

- Mentor: Cheng Chang

2 Introduction

In the very rapidly evolving field of natural language processing, transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) has revolutionized various aspects of language understanding and processing. BERT's approach, based on a 'masked language model' objective, enables it to capture a profound understanding of language context and nuances, and for it's embeddings and pre-trained weights to be used for finetuning for other downstream NLP tasks.

The primary goal of this project is twofold. Firstly, I implement a version of BERT (minBERT), designed to serve as a base model for further extensions. Secondly, I delve into utilizing minBERT and extending the model via finetuning for use in 3 downstream NLP tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. By leveraging BERT's sophisticated pre-trained context-aware representations, the project aims to investigate the synergies and performance enhancements that multi-task learning can introduce.

This approach posits that shared learning across these related yet distinct tasks can lead to a more versatile and robust model. My methodology encompasses a two-stage process: initial multi-task learning by interleaving samples from all 3 datasets for each task, followed by targeted fine-tuning on each task. This process is designed to capture the dual benefits of generalization from multi-task exposure and specificity from focused optimization. I also furthermore experimented with utilizing a learning rate scheduler for more efficient training. These techniques were instrumental in overcoming the often conflicting nature of task-specific optimizations in multi-task learning,

a challenge highlighted in studies like Yu et al.'s work on Gradient Surgery **?**. My results were promising – minBERT not only demonstrated strong performance across the tasks but also highlighted the potential of tailored fine-tuning strategies and learning rate adjustments in maximizing a language model's capabilities.

3 Related Work

In the field of natural language processing, two significant breakthroughs have laid the groundwork for contemporary advancements: the introduction of BERT (Bidirectional Encoder Representations from Transformers) by Devlin et al Devlin et al. (2019). and the Transformer model by Vaswani et al. BERT, brought a paradigm shift in NLP by enabling deep bidirectional representation learning, showcasing exceptional versatility across various NLP benchmarks and tasks. The latter, the Transformer model, eschewed traditional sequence learning methods for attention mechanisms, enabling more efficient and effective learning of language dependencies.

Various studies have expanded on BERT's utility in multi-task learning environments. One such research is the "BERT and PALs" paper Name (year), which introduces Projected Attention Layers (PALs) for efficient task adaptation. Similarly, the "Modeling Multi-task with Task Routing" paper proposes dynamic routing for tasks, optimizing the shared model's features while retaining task-specific characteristics.

Beyond these, the T5 model introduced by Raffel et al. unified diverse NLP tasks into a text-to-text framework, demonstrating the adaptability of BERT's architecture. Studies like Mosbach et al.'s work on fine-tuning nuances and Liu et al.'s on joint learning further highlight the intricate dynamics of adapting BERT for specific tasks. Dong et al.'s exploration of generalized autoregressive pretraining extends BERT's applications, underlining the model's extensive adaptability.

Together, these works form a comprehensive view of the evolving landscape of transformer-based models in NLP. They collectively show the impact of BERT and its derivatives, particularly in their adaptability to diverse and simultaneous NLP tasks, laying the foundation for this project's exploration of multi-task learning and fine-tuning strategies using minBERT.

4 Approach

4.1 minBert Architecture

The base BERT model architecture, as delineated in the original BERT paper Devlin et al. (2019), consists of 12 Encoder Transformer bidirectional layers, a hidden size of 768, and 12 self-attention heads. The Transformer layer in the BERT model encapsulates several key components: (a) a Multi-Head Self-Attention layer, followed by (b) an Additive and Normalization layer, (c) a Feed-Forward Layer and (d) a final Additive and Normalization Layer.

4.1.1 Multi-head Self-Attention

The Multi-Head Self-Attention mechanism, a key feature of the Transformer model from "Attention is All You Need" Vaswani et al. (2017), enables processing an input sequence through several attention heads in parallel. Each head independently computes attention scores, capturing the relevance of each word to others in the sequence.

For each head, attention scores are calculated using the formula below, where Q, K, and V stand for query, key, and value vectors, respectively, and d_k is the dimension of the key vectors:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

The outputs of all heads are concatenated and linearly transformed:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

with each $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$.

4.2 Sentiment Classification

A sentiment classifier which performs classification with the BERT pre-trained embeddings is implemented and fine-tuned on the SST and CFIMBD datasets. The Adam Optimizer, a method for efficient stochastic optimization is implemented and used in our classifiers.

In this part of the project, I implement a sentiment classifier that utilizes the pre-trained embeddings from BERT as the foundation for sentiment analysis.

The core of the classifier consists of a dropout layer and a linear layer. The dropout layer, is instrumental in reducing overfitting by randomly deactivating a fraction of neurons during training; leading to a more generalized model. Following the dropout layer is a linear layer, which maps the BERT embeddings to the sentiment labels.

For fine-tuning and optimizing this classifier, I leverage the Adam Optimizer, which is an iterative optimization algorithm used to minimize the loss function during the training of neural networks

The fine-tuning process involves two datasets: the Stanford Sentiment Treebank (SST) and the CFIMBD dataset. The classifier is fine-tuned on the SST dataset, which is a standard benchmark in sentiment analysis and offers a diverse range of movie reviews. The classifier is also separately fine-tuned on the CFIMBD dataset, which also consists of highly polar movie reviews.

4.3 Multi-task Classification

A multi-task classifier which can simultaneously perform 3 tasks is implemented by extending and fine-tuning on top of the pre-trained BERT model.

1. **Sentiment Analysis:** This task involves classifying the polarity of a text, determining whether it is positive, somewhat positive, negative, somewhat negative, or neutral.
   - *Example:*
     - **Input:** "Light, silly, photographed with colour and depth, and rather a good time."
     - **Output:** Positive (Sentiment: 4)
2. **Paraphrase Detection:** The goal here is to identify if two given texts are paraphrases of each other, meaning they essentially convey the same message.
   - *Example:*
     - **Input 1:** "What is the step by step guide to invest in share market in india?"
     - **Input 2:** "What is the step by step guide to invest in share market?"
     - **Output:** Is paraphrase: No
3. **Semantic Textual Similarity:** This task measures the degree of semantic equivalence between two sequences of words on a scale from 0 (not at all related) to 5 (same meaning).
   - *Example:*
     - **Input 1:** "The woman is playing the violin."
     - **Input 2:** "The young lady enjoys listening to the guitar."
     - **Output:** Similarity Score: 0

The approach that I followed to achieve a robust multi-task classifier based on BERT, my approach was structured into two main phases: initial multi-task training and subsequent dataset-specific fine-tuning. During the multi-task learning, for each training epoch I interleaved samples from each of the 3 datasets (SST, Quora, and SemEval).

4.3.1 Multi-task Training

The first phase involved multi-task training, where the pre-trained BERT model was adapted to simultaneously handle three diverse NLP tasks: sentiment analysis (SST), paraphrase detection (Quora), and semantic textual similarity (SemEval). A key aspect of this phase was interleaving of samples from each dataset during each training epoch. This interleaving was crucial, especially given the significant size disparity among the datasets, with the Quora dataset being substantially larger. To manage this disparity and ensure balanced exposure to each task, I used a proportional sampling strategy. The number of batches from the Quora dataset in each epoch was determined by

the ratio of the sizes of the SST and Quora datasets. During each epoch, I cycled through the SST and SemEval datasets while simultaneously iterating through the calculated number of Quora batches. This interleaving meant that in each epoch, the model trained on a balanced mix of data from all three tasks, enhancing its ability to generalize across them.

$$\text{quora\_batches\_per\_epoch} = \left\lfloor \frac{\text{len(sst\_train\_dataloader)} \times \text{len(sst\_train\_data)}}{\text{len(para\_train\_data)}} \right\rfloor \tag{1}$$

During the multi-task training, if there was improvement in one of the 3 tasks, then I updated and saved the model.

A significant concept in multi-task learning involves the aggregation of losses from different tasks, as discussed in the paper "Gradient Surgery for Multi-Task Learning" Yu et al. (2020). In the implementation by Bi et al. Yu et al. (2020) demonstrates this by summing up the losses from different tasks, such as category classification and named entity recognition. By following this approach, I calculate the total loss by:

$$L_{\text{total}} = L_{\text{sentiment\_analysis}} + L_{\text{text\_similarity}} + L_{\text{paraphrase}} \tag{2}$$

4.3.2 Additional Single-task Finetuning

Following the multi-task training, the model underwent additional fine-tuning phases for each dataset separately. This fine-tuning was critical to hone the model's capabilities for each specific task, allowing for task-specific optimizations and adjustments. The fine-tuning on individual datasets addressed any task-specific nuances that might not have been fully captured during the multi-task learning phase.

During the individual fine-tuning phase of my project, in order to ensure that the performance on each task did not degrade while optimizing for specific datasets, I used the following approach:

For each of the three tasks - I first determined a baseline accuracy from the multi-task training phase, which was used for setting a a performance threshold for the additional fine-tuning phase.

$$\text{threshold}_{\text{task}} = 0.90 \times \text{accuracy}_{\text{multi-task, task}}$$

The fine-tuning process for each dataset was then conducted with this threshold in mind. The key principle was that the model's accuracy on a specific task, post fine-tuning, must exceed the established threshold for the updates to be retained. This ensured that fine-tuning led to actual improvements rather than detrimental overfitting or deviation for any of the tasks.

This structured approach – starting with multi-task training to build a strong, versatile foundation and following up with targeted fine-tuning – was instrumental in achieving a classifier that excelled in handling a variety of NLP challenges.

I also experimented with a learning rate step scheduler. The step scheduler was used to adjust the learning rate at regular intervals. The rationale behind using a step scheduler was to methodically reduce the learning rate as the training progressed, thereby fine-tuning the model's parameters more delicately as it approached optimal performance.

5 Experiments

The datasets used where as follows:

5.1 Data

For the sentiment classification task using a single task classifier, two primary datasets were used:

1. **The SST Dataset (Stanford Sentiment Treebank):**
    - **Overview:** The dataset includes 11,855 sentences extracted from movie reviews. In addition to these sentences, it comprises 215,184 unique phrases.

Dataset	Size	Split (train/dev/test)	Evaluation Metric	Task
Quora	400,000	141,506 / 20,215 / 40,431	Accuracy	Paraphrase detection
SemEval	8,628	6,041 / 864 / 1,726	Pearson Correlation	Semantic text analysis
SST	11,855	8,544 / 1,101 / 2,210	Accuracy	Sentiment classification
CFIMDB	2,434	1,701 / 245 / 488	Accuracy	Sentiment classification

Table 1: Datasets

- **Labels:** The phrases are annotated with sentiment labels including 'negative', 'somewhat negative', 'neutral', 'somewhat positive', and 'positive'.

2. **The CFIMBD Dataset:**
   - **Overview:** This dataset contains 2,434 movie reviews.
   - **Labels:** Each review is labeled as either 'positive' or 'negative'.

For the multi-task classification task, 3 datasets were used:

For the multi-task classification task, three distinct datasets were utilized for different NLP tasks:

1. **Sentiment Analysis (SST Dataset):**
   - **Purpose:** Employed for sentiment analysis, involving classifying the sentiment polarity in movie review sentences.
   - **Dataset Description:** The Stanford Sentiment Treebank (SST) includes a rich collection of sentences and phrases from movie reviews.
2. **Paraphrase Detection (Quora Dataset):**
   - **Purpose:** Used for detecting paraphrases among question pairs.
   - **Overview:** Contains 400,000 question pairs with annotations indicating whether the pairs are paraphrases of each other.
3. **Semantic Textual Analysis (SemEval Dataset):**
   - **Purpose:** Applied for measuring the degree of semantic similarity between sentence pairs.
   - **Overview:** Consists of 8,628 pairs of sentences with varied levels of semantic similarity.

5.2 Evaluation method

Describe the evaluation metric(s) you use, plus any other details necessary to understand your evaluation. Some projects will have clear metrics from prior work on given datasets, but we realize that other projects will define their own metrics. If you're defining your own metrics, be clear as to what you're hoping to measure with each evaluation method (whether quantitative or qualitative, automatic or human-defined!), and how it's defined.

5.3 Experimental details

In this study, a series of experiments were conducted to evaluate the effectiveness of various training strategies for the multi-task classifier. These experiments included:

1. **Multi-Task Training:**
   - The first experiment involved training the classifier across multiple NLP tasks simultaneously. The focus was to assess the ability of the model to generalize knowledge from the pre-trained BERT model across different tasks.
2. **Multi-Task Training with Additional Fine-Tuning:**
   - Building on the initial multi-task training, this phase incorporated additional fine-tuning for each specific task and dataset. The goal was to enhance the model's task-specific performance, building upon the generalized capabilities acquired in the first experiment.
3. **Multi-Task Training with Additional Fine-Tuning and Step Scheduler:**

- This experiment extended the previous approach by integrating a step scheduler to adjust the learning rate at predefined intervals. The inclusion of the step scheduler aimed to refine the fine-tuning process, targeting more precise and effective adjustments to the model's training.

The purpose of these experiments was to iteratively build upon the insights gained from each phase, progressively improving the classifier's capabilities for multi-task learning in NLP and identifying the most effective training strategies.

5.4 Results

Below are the results from the initial experiments with the single task sentiment classifier and the multi-task classifier which was originally finetuned just on the SST dataset without any extensions.

Classifier Type	Dataset	Pretraining Accuracy	Finetuning Accuracy
Sentiment Classifier	SST	0.390	0.517
Sentiment Classifier	CFIMDB	0.780	0.967
Multi-Task Classifier	SST	0.465	0.375
Multi-Task Classifier	Quora	0.510	0.574

Table 2: Performance of Sentiment and Multi-Task Classifier

Dataset	Pretraining Correlation	Finetuning Correlation
SemEval	-0.094	-0.125

Table 3: Performance on SemEval

Below are the results from the multi-task classification experiments:

Dataset	Multi-Task Training	+ Finetune on All Datasets	+ Finetune + Step
SST Accuracy	0.510	0.504	0.496
Paraphrase Accuracy	0.574	0.657	0.697
STS Dev Correlation	-0.125	0.355	0.625

Table 4: Performance Across Different Training Stages

Results on the test datasets:

6 Analysis

7 Analysis

7.1 Multi-Task Training

The initial multi-task training phase showed promising results, especially in the context of task adaptability. The model demonstrated a capability to handle different types of tasks, although with varying degrees of success. For instance, in sentiment analysis (SST), the accuracy achieved was 0.510, indicating a reasonable understanding of sentiment in text. However, in the semantic textual similarity task (STS), a negative correlation was observed, suggesting difficulties in accurately capturing and comparing semantic nuances between sentence pairs.

7.2 Fine-Tuning on Individual Tasks

The addition of fine-tuning for each specific dataset post multi-task training brought notable improvements. This phase was crucial in tailoring the model more closely to the intricacies of each task. The improvements in accuracy for SST and paraphrase detection after fine-tuning indicate that the model benefited from the focused training, adapting better to the specific requirements of these tasks.

Dataset	Test Accuracy/Correlation	Change from Previous
SST Test Accuracy	0.489	+0.011
Paraphrase Test Accuracy	0.701	+0.008
STS Test Correlation	0.342	-0.016

Table 5: Test Performance and Changes from Previous Stage

7.3 Integration of Step Scheduler

The incorporation of a step scheduler in the training regime had a significant impact, especially observed in the increase of paraphrase detection accuracy and the STS correlation. The gradual refinement of learning rates likely contributed to more nuanced adjustments in the model's weights, leading to a better grasp of complex task-specific features.

7.4 Overall Implications

Across the various stages of training and fine-tuning, the model exhibited an evolving understanding of each task. The continuous improvements, especially in paraphrase detection and semantic textual similarity, underscore the effectiveness of a multi-phase training approach. However, the nuanced decrease in SST accuracy through these stages also points to the challenges in balancing multi-task learning with task-specific optimization.

In summary, my qualitative analysis reveals that while the multi-task training model shows adaptability and potential for learning across diverse tasks, the fine-tuning phases are pivotal in honing task-specific competencies. The varying degrees of success across different tasks and stages highlight the complex interplay between generalizability and specialization in NLP models.

8 Conclusion

This project embarked on an ambitious journey to explore the capabilities of BERT in a multi-task setting and further refine its performance through targeted fine-tuning. The key findings of our experiments are:

- The implementation of a multi-task classifier based on BERT, which showed proficiency in handling various NLP tasks with differing requirements.
- The significant enhancements in model performance, particularly in paraphrase detection and semantic textual similarity, achieved through additional fine-tuning and step scheduler integration.
- The challenges in maintaining a balance between general task adaptability and specialized task optimization.

Our work sheds light on the intricate dynamics of multi-task learning and fine-tuning in NLP. Despite the advancements made, limitations such as slight declines in certain task performances remind us of the complexities inherent in this domain. Future work could explore more sophisticated balancing mechanisms in multi-task learning and further investigate fine-tuning techniques to bolster model performance across all tasks.

This project not only advances our understanding of BERT's applications in multi-task environments but also opens pathways for future explorations in the evolving landscape of NLP.

9 Conclusion

This project represents a comprehensive exploration into the domain of natural language processing using a minimal version of BERT, followed by an extension to multi-task classification. My main findings reveal that:

- The implementation of minBERT, while computationally efficient, effectively leveraged the capabilities of the original BERT model in understanding and processing language.

- The multi-task classifier demonstrated versatility and robustness across various NLP tasks, including sentiment analysis, paraphrase detection, and semantic textual similarity.
- The addition of fine-tuning on individual datasets, complemented by the integration of a step scheduler, enhanced task-specific performances.

Through this project, I have gained valuable insights into the complexities of multi-task learning and the effectiveness of fine-tuning strategies in improving model performance. However, there are limitations, such as the potential for overfitting on specific tasks and the challenge of balancing performance across different tasks.

For future work, several avenues present themselves:

- Exploring more advanced techniques for balancing multi-task learning, possibly through dynamic weighting of task losses or gradient normalization methods.
- Investigating alternative fine-tuning approaches to further improve model generalization and prevent task-specific overfitting.
- Extending the model's application to additional NLP tasks and experimenting with larger, more diverse datasets to test its scalability and adaptability.

In conclusion, this project has not only contributed to my understanding of BERT and its applications in multi-task learning but also opened up new possibilities for future research in the field of NLP.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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. Association for Computational Linguistics.

Author's Name. year. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. *Journal Name*, vol(num):pages.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010. Curran Associates Inc.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Name of Journal, if available*, vol(num):pages.