Multi-task Learning for BERT

Stanford CS224N Default Project

Xinyan He
Department of Computer Science
Stanford University
xinyanh@stanford.edu

Wei Zhao
Department of Computer Science
Stanford University
wzhao18@stanford.edu

Abstract

The pretrain-finetune paradigm has demonstrated remarkable efficacy in various natural language processing tasks. However, while this method has proven effective in enhancing single-task performance, achieving robust performance in multi-task settings presents a set of distinct challenges, such as single-task overfitting and catastrophic forgetting. In this study, we examine the multi-task performance of pretrained language models on three downstream tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. We employ BERT as our base model and incorporate a variety of optimization techniques, including interleaved training and gradient surgery, to improve the model's generalization capabilities. Our evaluation demonstrates significant improvements achieved by our proposed methods, elevating the collective performance from a baseline score of 0.647 to 0.787. Furthermore, we conduct a comparative analysis between the performance of single-task and multi-task fine-tuning, highlighting the effectiveness of multi-task learning in improving performance of language models by leveraging shared representations across tasks.

1 Key Information to include

The mentor of this project is Jingwen Wu. There is no external collaborator or mentor.

2 Introduction

In recent years, the advent of transfer learning has profoundly transformed the realm of natural language processing (NLP) (Ruder et al., 2019). This paradigm shift is epitomized by the two-stage process known as the pretrain-finetune approach, which first leverages unsupervised learning to train models with vast amount of text corpora, enabling them to learn intricate linguistic structures, nuanced sentence semantics, and capture the complex interplay between words and their contexts. Subsequently, these pretrained models can be fine-tuned utilizing comparatively small, task-specific datasets, allowing them to efficiently adapt their sophisticated knowledge to specific downstream applications and render robust performance (Devlin et al., 2019; Gao et al., 2022). Despite these advances, training these models for multi-task settings faces numerous challenges. First, imbalanced multi-task fine-tuning can lead to *single-task overfitting*, where the model becomes overly specialized in one task at the expense of others, compromising its ability to generalize effectively across multiple domains (Crawshaw, 2020). Another significant hurdle is the issue of *catastrophic forgetting*, where the model's performance on previously learned tasks deteriorates as it is fine-tuned on new tasks (Serra et al., 2018). Moreover, parameter updates for multiple objectives can suffer from *gradient conflict*, hindering the progress of training and potentially leading to suboptimal convergence (Yu et al., 2020).

In this work, we explore methods to overcome the challenges in multi-task learning through a case study that focuses on the fine-tuning of the pretrained BERT model (Devlin et al., 2019) for three downstream tasks: Sentiment Analysis (SA), Paraphrase Detection (PD), and Semantic Textual

Similarity (STS). Our implementation features an additional pretraining stage, which leverages contrastive learning-based supervised and unsupervised learning objectives (Chen et al., 2020; Gao et al., 2022) to adapt the pretrained BERT model with the domain-specific data in our study. To combat the potential of single-task overfitting and catastrophic forgetting, we incorporate a novel interleaved training strategy coupled with circular data sampling, ensuring balanced learning across different tasks. We also explore numerous approaches to mitigate conflicting parameter updates, including adversarial degularization (Jiang et al., 2020), gradient surgery (Yu et al., 2020), and gradient clipping (Zhang et al., 2020). Furthermore, we leverage Low-rank Adaptation (LoRA) (Hu et al., 2021) to train multiple task-specific models with specialization in individual tasks using reduced computational resources. Our evaluation demonstrates significant improvements from our proposed methods, elevating the collective score of the three tasks from 0.647 to 0.787. Moreover, our experiments show that multi-task learning on the same model surpasses the performance of training each task on a separate model, highlighting the potential of multi-task learning in enhancing the performance and generalization capabilities of language models.

3 Related Work

Language Pretraining. Learning universal language representations through pretraining has significantly enhanced performance across numerous tasks in NLP, including natural language inference, named entity recognition, and question answering (Devlin et al., 2019). Various pretraining objectives have been proposed, each designed to capture different aspects of language understanding. One notable example is the Masked Language Model (MLM), employed by BERT (Devlin et al., 2019). In this approach, random tokens within a sentence are obscured and the model is tasked with inferring these hidden words based solely on the surrounding context, thereby training the model to develop a deep understanding of the relationships between words and their contexts. Another prominent technique in representation learning is contrastive learning, exemplified by SimCLR (Chen et al., 2020) and SimCSE (Gao et al., 2022), which maps semantically similar examples closer in the embedding space and pull dissimilar ones further apart. This objective effectively captures the locality between semantically similar sentences and helps the model learn more meaningful and robust representations.

Multi-task Learning. Multi-task learning is a training paradigm in which models are trained to perform multiple tasks simultaneously. However, this approach introduces distinct challenges that require specialized optimizations to address effectively, as demonstrated by Crawshaw (2020). One crucial optimization technique in multi-task learning is the use of regularization. For instance, Jiang et al. (2020) propose the SMART learning framework, which leverages Adversarial Regularization combined with Momentum Bregman Proximal Point Optimization to prevent models from overfitting the training data failing to effectively generalize. Another promising direction of optimization in multi-task learning is Gradient Surgery, exemplified by Gradient Episodic Memory (Lopez-Paz and Ranzato, 2022) and PCGrad (Yu et al., 2020). The main idea behind these works is to replace a task gradient that conflicts with another by a modified version that minimizes the conflict. For instance, PCGrad achieves this by projecting a task gradient onto the normal plane of the conflicting gradient.

Parameter-efficient Fine-tuning. Parameter-efficient fine-tuning techniques have gained significant attention in the field of deep learning, particularly in light of the growing scale of model sizes and associated computational cost. Among these techniques, Low-Rank Adaptation (LoRA) (Hu et al., 2021) has emerged as a popular approach that significantly reduces the computational and memory requirements of fine-tuning by injecting trainable rank decomposition matrices into specific layers of a pre-trained model. Specifically, for each selected layer of the model, LoRA introduces low-rank matrices that decompose the original weight matrix, allowing for more efficient fine-tuning. During the fine-tuning phase, the original weight matrices of the model remain unchanged, and instead, the outputs from the low-rank matrices are used to adjust the activations of the layers in the model, providing a more computationally efficient means of adapting the model to new tasks.

4 Approach

In the multi-task setting of our study, the proposed model is required to perform three distinct tasks simultaneously: (1) Sentiment Analysis, which involves classifying the polarity of a given text, i.e., determining whether the expressed opinion in the text is positive, negative, or neutral; (2) Paraphrase Detection, which entails ascertaining whether two given sentences convey the same meaning and

can be considered paraphrases of each other; and (3) Semantic Textual Similarity, which requires the model to assign a numerical score to quantify the degree of semantic similarity between two sentences, with higher scores indicating greater similarity in meaning.

Baseline. For our baseline implementation, we employ the $\text{BERT}_{\text{base}}$ model (Devlin et al., 2019) with pretrained weights. Given that the inputs for the three tasks consist of one or more sentences, we utilize the embedding of the first token ([CLS]) as the representative embedding for each sentence, as it is designed to capture the overall meaning of the input. Our initial approach involves using pretrained BERT to produce an embedding for each input sentence separately. Depending on the specific task, the output embeddings are then concatenated and passed through a corresponding linear layer to produce the final output logits, allowing the model to make task-specific predictions. Regarding the choice of loss functions, we employ cross entropy for sentiment analysis, as it is well-suited for multi-class classification tasks; binary cross entropy for paraphrase detection, given its binary classification nature; and Mean-Squared Error (MSE) for semantic textual similarity, as the similarity scores are continuous numerical values. In each epoch, training is performed sequentially over the three datasets.

Cross Encoding. We speculate that the design of the baseline method, wherein the final linear layer are solely responsible for mapping the concatenated embeddings to the output logits, may impose limitations on the model's performance. To address this potential drawback, we experiment with the cross encoding technique, which involves concatenating the input sentences upfront and utilizing the combined input to generate a unified embedding. Specifically, we concatenate both the input ids and the attention masks of the two sentences, with the former separated by the [SEP] token and the latter by a mask of 1. By employing this technique, the model is able to consider the interaction between the sentences from the initial stages of processing, enabling it to extract features that are more indicative of the underlying semantic relationships.

Interleaved Training with Sampling. In our initial approach, the training process iterates over the individual datasets of the three tasks sequentially, which we speculate could lead to the later training steps overwriting the updates from previous tasks. This problem is further exacerbated by the significant imbalance in training dataset sizes among the three tasks (for instance, 8,543 and 283,012 training samples for SA and PD, respectively), causing a notable disparity in the learning process. To address these potential issues, we adopt an interleaved training approach that equalizes the number of batches based on the largest dataset. For each epoch, we sample an equivalent amount of data from the smaller datasets to match the number of batches in the largest dataset. This method aims to ensure balanced learning across tasks, thereby enhancing the model's ability to generalize effectively across multiple tasks without any one task dominating the training process due to its larger dataset size. We also set the new loss function as the sum of the individual task losses, as

$$\mathcal{L} = \mathcal{L}_{\text{sa}} + \mathcal{L}_{\text{pd}} + \mathcal{L}_{\text{sts}} \tag{1}$$

, where $\mathcal{L}_{\text{sa}}$, $\mathcal{L}_{\text{pd}}$, and $\mathcal{L}_{\text{sts}}$ represent the losses for sentiment analysis, paraphrase detection, and semantic textual similarity, respectively. This interleaved training strategy not only ensures a more balanced representation of each task during the training process but also proves to be computationally more efficient, as it allows for a more consistent and synchronized update of the model's parameters across all tasks.

Contrastive Learning. In our exploration of enhancing the model's performance, we have experimented with an additional pretraining step through the use of contrastive learning (Chen et al., 2020), which aims to map semantically similar examples (positive pairs) closer in the embedding space while pulling dissimilar ones (negative pairs) further apart. For a batch of data of size N, the loss function with respect to a positive pair (x_i, x_i^+) is defined as

$$l_i = -log \frac{e^{\text{sim}(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(h_j, h_j^+)/\tau}}, \tag{2}$$

where τ is the temperature hyperparameter, h_i and h_i^+ are the generated embeddings of x_i and x_i^+ respectively, and sim refers to the cosine similarity function $\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$. Our implementation of contrastive learning-based pretraining consists of both a supervised and unsupervised stage. Inspired by SimCSE (Gao et al., 2022), we utilize dropout as a noise mechanism to generate different but semantically similar pairs of sentences as positive examples, with negative pairs being the different sentences from the same batch. This approach allows us to perform contrastive learning in an unsupervised manner.

Moreover, we have discovered that many sentence pairs in the PD and STS datasets exhibit semantic similarity with ground truth verification. For instance, if two sentences are labeled as paraphrases of one another, they can be naturally defined as a positive pair. Leveraging this insight, we filter out positive input pairs from the PD and STS datasets based on their labels and use them for contrastive learning. By incorporating this additional pretraining step, we aim to enhance the model's ability to capture the semantic relationships between sentences, ultimately leading to improved performance across all three tasks in the multi-task learning framework.

Gradient Surgery. A critical challenge in multi-task learning is that the directions of gradient for different tasks may conflict with one another, thereby making it difficult for the model to effectively learn multiple tasks simultaneously (Crawshaw, 2020). To mitigate this potential problem, we have explored integrating the gradient surgery technique proposed by Yu et al. (2020) into our model. This method resolves conflicts between gradients by subtracting a gradient by its projection onto the gradient of another task, as illustrated in Equation (3), where $\mathbf{g}_i$ and $\mathbf{g}_j$ are the gradients of tasks i and j respectively. We apply gradient surgery on the parameter updates within the BERT module, as this module is shared across all three tasks. Specifically, within each batch during training, we update the gradient for each task by projecting it onto the normal plane of the other two tasks and performing the subtraction. By aligning the gradients in this manner, we aim to promote more harmonious updates during the training process, ensuring that the gradients for each task interfere minimally with the others.

$$\mathbf{g}_i = \mathbf{g}_i - \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2} \cdot \mathbf{g}_j \tag{3}$$

Adversarial Regularization. In an effort to enhance the generalization capability of our multi-task learning framework, we have experimented with incorporating the Smoothness-Inducing Adversarial Regularization technique, inspired by the SMART learning framework (Jiang et al., 2020). This regularization introduces an additional term $\lambda_s \mathcal{R}_s(\theta)$ to the standard task-specific training loss $\mathcal{L}(\theta)$, where $\lambda_s > 0$ is a tuning parameter that specifies the weight of the regularization term, and $\mathcal{R}s(\theta)$ is defined as

$$\mathcal{R}_s(\theta) = \frac{1}{n} \sum_{i=1}^{n} \max_{\|\tilde{x}_i - x_i\|_p \leq \epsilon} \ell_s(f(\tilde{x}_i; \theta), f(x_i; \theta)), \tag{4}$$

with $\epsilon > 0$ being a tuning parameter. The choice of the loss function ℓ_s depends on the task type. Following the original choice in SMART, we opt for the symmetric Kullback-Leibler divergence loss (Joyce, 2011) for classification tasks and the MSE for regression tasks. The adversarial nature of the regularization encourages the model to learn features that are less sensitive to small perturbations in the input, thereby improving its ability to generalize well to unseen data.

Gradient Clipping. Gradient clipping is a widely employed technique in the training of deep neural networks, serving as a complementary method to standard gradient descent. The objective of gradient clipping is to limit the norm of the gradient to a predefined value $c > 0$ at every iteration of the training process. This is achieved by scaling the gradients by

$$\mathbf{g} \leftarrow \mathbf{g} \cdot \frac{c}{\|\mathbf{g}\|}, \tag{5}$$

where $\mathbf{g}$ represents the gradient vector. By restricting the magnitude of the gradients, gradient clipping effectively mitigates the potential issue of gradient explosion, a phenomenon particularly prevalent in recurrent neural networks, as elucidated in (Goodfellow et al., 2016). In the context of our multi-task learning setting, gradient clipping helps prevent aggressive updates for individual tasks and ensures that the updates for each task are balanced and controlled, thereby promoting a more stable and effective learning process.

Mixture of Low-Rank Experts. In multi-task settings, fine-tuning a dedicated model for each specific task can effectively train the model to specialize in that task and potentially maximize performance. However, the fatal downside of this approach is the requirement of a separate model for each task, which significantly increases the number of trainable parameters and training time. To mitigate these challenges while still leveraging the benefits of specialized fine-tuning, we introduce the Mixture of Low-rank Experts, a novel approach that utilizes Low-Rank Adaptation (LoRA) to efficiently fine-tune a task-specific model for each task. Here we will update the weight of corresponding layer into:

$$\mathbf{W} = \mathbf{W_0} + \mathbf{BA} \tag{6}$$

where W_0 represents the weight matrix obtained from pre-training, and A and B are the parameters introduced during fine-tuning. Specifically, in our model we will decompose three weight matrices in each BERT layer - $interm\_dense$, $attention\_dense$, and out_dense - into two low-rank matrices each. This decomposition allows for more efficient fine-tuning, as the model only needs to update the parameters in the low-rank matrices. The approach enables the model to maximize the performance in each task while maintaining a manageable number of trainable parameters and avoiding a prolonged training time.

5 Experiments

Data. In this study, we employ three datasets corresponding to the three tasks under investigation. For sentiment analysis, we employ the Stanford Sentiment Treebank (SST-5) (Socher et al., 2013), which contains 11,855 single sentences extracted from movie reviews. Each phrase in the dataset is labeled as negative, somewhat negative, neutral, somewhat positive, or positive. For paraphrase detection, we use the Quora Question Pairs (QQP) (Fernando and Stevenson, 2008) dataset, which consists of 404,304 pairs of sentences. Each pair is labeled as either a paraphrase of each other or not. Finally, for semantic textual similarity, we utilize the Semantic Textual Similarity Benchmark (STS-B) (Agirre et al., 2013), which contains 8,631 sentence pairs. Each pair in the STS-B dataset is assigned a similarity score between 0 and 5. Each dataset is divided into TRAIN, DEV and TEST subsets.

Evaluation method. For sentiment analysis and paraphrase detection, we utilize prediction accuracy as the primary evaluation metric. In the case of the semantic textual similarity task, we employ the Pearson correlation coefficient as the evaluation metric. This measure assesses the linear relationship between the predicted similarity scores and the ground truth scores, with higher values indicating a stronger positive correlation and hence better performance in capturing the semantic similarity between sentence pairs. we normalize the correlation coefficient from its original range of -1 to 1 to a range of 0 to 1 as its score. The overall score for a model is measured by the collective performance of the three tasks, as illustrated by Equation (7).

$$\text{score} = \frac{1}{3} \times \text{Accuracy}_{\text{SA}} + \frac{1}{3} \times \text{Accuracy}_{\text{PD}} + \frac{1}{3} \times \frac{(\text{Correlation}_{\text{STS}} + 1)}{2} \tag{7}$$

Experimental details. In our experimental setup, we implement the BERT architecture and utilize pretrained weights identified by "google-bert/bert-base-uncased" from the Hugging Face Transformers library. To ensure a fair comparison between the various techniques, we maintain a consistent set of hyperparameters during the model exploration stage, employing a batch size of 8, a hidden dimension of 768, a dropout rate of 0.3 for the hidden layers, and a learning rate of 1e-5 with full model fine-tuning. For optimization, we use the default Adam optimizer configuration with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and weight decay = 0. For Mixture of Low-Rank Experts with LoRA, we set the rank of matrices as 256 and $alpha = 32$. All models are fine-tuned with the entire model without parameter freezing. We train each model with 10 epochs, as we observe the loss typically converges before reaching this point. We select the checkpoint with the highest overall score on the DEV dataset for each model. Once the best-performing model is identified based on the initial experiments, we conduct an additional hyperparameter tuning stage to further refine its performance.

Results. In Table 1, we present a comprehensive comparative analysis of the performance metrics achieved by various methods across the three tasks: sentiment analysis (SA), paraphrase detection (PD), and semantic textual similarity (STS). The cross encoding technique, which involves concatenating input sentences upfront to generate a unified embedding, significantly improves the baseline performance, particularly for the STS correlation score, resulting in a substantial increase from 0.374 to 0.881. Further enhancement is observed when the interleaved training strategy coupled with sampling is employed, leading to an overall score improvement of 1.7, reaching 0.778. This improvement demonstrates the effectiveness of ensuring balanced learning across the tasks. Building upon the foundation of cross encoding and interleaved training, we investigate the impact of applying several techniques, among which adversarial regularization, contrastive learning, and gradient clipping result in improvement of the overall score, with scores of 0.781, 0.783, and 0.787, respectively. Gradient surgery, with a score of 0.777, does not demonstrate effectiveness in this setting. Notably, gradient clipping yields the most significant performance improvement, elevating the score to 0.787. Additionally, we explore the combination of contrastive learning-based pretraining with gradient

Model	# Trainable Parameters	SA Acc.	PD Acc.	STS Corr.	Overall Score
Baseline	112M	0.450	0.805	0.374	0.647
CE	112M	0.453	0.889	0.881	0.761
CE + Interleaved	112M	0.496	0.898	0.878	0.778
CE + Interleaved + Surgery	112M	0.495	0.894	0.883	0.777
CE + Interleaved + Regularization	112M	0.500	0.900	0.883	0.781
CE + Interleaved + Contrastive	112M	0.509	0.900	0.883	0.783
CE + Interleaved + Clipping	112M	0.518	**0.901**	**0.883**	**0.787**
CE + Interleaved + Clipping + Contrastive	112M	**0.521**	0.901	0.873	0.786
CE + Mixture + Clipping	336M	0.515	0.904	0.877	0.786
CE + Mixture + LoRA + Clipping	81M	0.499	0.886	0.846	0.769
Best-model Test Performance		0.523	0.901	0.879	0.788

Table 1: Performance comparison of various models on the DEV dataset and the best-performing model's evaluation on the TEST dataset. The models under consideration include Cross Encoding (CE); Interleaved Training with Sampling (Interleaved); Gradient Surgery (Surgery); Adversarial Regularization (Regularization); Gradient Clipping (Clipping); Contrastive Learning (Contrastive); and Mixture of Low-Rank Experts (Mixture).

Learning Rate	SA	PD	STS
1e-3	0.253	0.631	0.040
1e-4	0.402	0.832	0.822
1e-5	**0.518**	**0.901**	**0.883**
5e-5	0.458	0.873	0.868
1e-6	0.504	0.888	0.878

Dropout Rate	SA	PD	STS
0.1	0.505	0.901	0.882
0.2	0.517	0.901	0.883
0.3	**0.518**	**0.901**	**0.883**

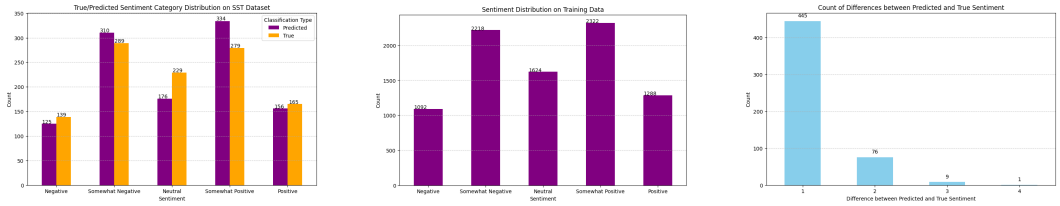
Table 2: Best-model performance with various learning rates and dropout rates

clipping during the fine-tuning stage; however, the performance does not surpass that of gradient clipping alone. In the hyperparameter tuning stage, presented in Table 2, we tested the performance of the best model (CE + Interleaved + Clipping) for different learning rates and dropout rates, finding that the default hyperparameters with a learning rate of 1e-5 and a dropout rate of 0.3 result in the best performance. Finally, we applied our best model, validated on the DEV dataset, to the TEST dataset and achieved a score of 0.788, which essentially maintains the same level as the score on the DEV dataset, demonstrating the generalization capability of our model.

We also examine the performance of training each task individually. Interestingly, paraphrase detection (PD) achieves a higher single-task score of 0.904 compared to the multi-task learning score of 0.901. For sentiment analysis (SA) and semantic textual similarity (STS), both scores are lower than those obtained through multi-task learning. We attribute this observation to the smaller dataset sizes of SST-5 and STS-B for SA and STS, respectively, compared to the larger QQP dataset for PD. In the multi-task learning setting, both SA and STS benefit from the knowledge learned from the QQP dataset, resulting in improved performance. However, since PD already has the largest dataset to learn from, it does not experience a substantial benefit from multi-task learning. The number of trainable parameters in the single-task learning objective is 336M, triple that of the multi-task learning paradigm. To reduce the number of trainable parameters, we employ Low-Rank Adaptation (LoRA), resulting in 81M parameters, which is smaller than that of the base BERT model. However, the performance after applying LoRA drops noticeably, achieving only 0.769. This observation highlights the effectiveness of multi-task learning, which leverages a single model to achieve more robust performance compared to training individual models separately, while also being more parameter-efficient.

6 Analysis

Sentiment Analysis. Figure 1a shows the distribution of our model's predictions of the sentiment classes of the DEV dataset compared to the ground truth labels. While the model achieves high



(a) Sentiment classification distribution

(b) Sententiment class distribution of training dataset

(c) Distance between predicted and true sentiment classes

Figure 1: Model performance decomposition for Sentiment Analysis.

Sample	True	Predicted	Characteristics
"I did not dislike the film, but it was far from great."	2	3	Negation
"Not only unfunny, but downright repellent."	0	3	Negation
"Oh great, another movie about superheroes. Just what we needed."	2	4	Irony or Sarcasm

Table 3: Failure cases for sentiment analysis

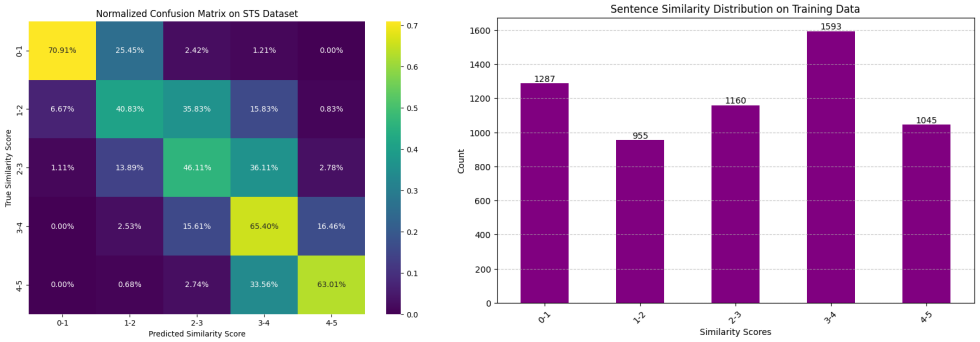
accuracy in the positive and negative cases, the error rate in the intermediate sentiment classes, namely somewhat negative, somewhat positive, and neutral categories, is noticeably higher. Specifically, the model predicts more somewhat negative and somewhat positive instances than the true labels, while predicting fewer in the neutral class. We attribute this observation to two contributing factors: firstly, the imbalanced training data distribution illustrated in 1c, which shows significantly more samples in somewhat negative and somewhat positive, likely causing the model to become biased towards these categories; and secondly, the model's suboptimal performance on sentences with more complicated semantics, such as double negatives and sarcasm. Table 3 presents two examples where the model generates incorrect predictions. The first sentence expresses a double negative tone with the phrase "did not dislike," while the second sentence employs sarcasm. These complex linguistic phenomena pose challenges for the model in accurately capturing the intended sentiment. Despite these limitations, Figure 1b reveals that the majority of the misclassified instances are only one class away from the true label, indicating that the model possesses a reasonable level of understanding, even in cases where it fails to predict the exact sentiment class.

Paraphrase Detection. Our model overall exhibits commendable performance for the paraphrase detection task, achieving an accuracy of 90.1% with a false positive rate of 8.67% and a false negative rate of 11.72%. Upon analyzing the failed cases, we observed that the model tends to predict "true" for sentence pairs that share a significant number of identical words, even when these pairs often express different or opposite meanings. Table 3 presents several examples with similar sentence wordings and structures that mislead the model into believing they are paraphrases of each other. This observation suggests that the model may have a higher reliance on word-level matching rather than comprehending the underlying semantics and nuances of the sentences. Addressing this limitation may requiring further scaling of the model size or using more effective learning objectives that encourages the model to distinguish similar sentences with different semantic meaning.

Semantic Textual Similarity. Figure 2a visualizes our model's predictions compared to the ground truth using a confusion matrix, with similarity scores divided into 5 buckets of width 1. Ideal predictions would result in a high density along the diagonal axis of the confusion matrix. Our model performs well for scores between 0-1, 3-4, and 4-5, with over 60% of predictions matching the ground truth labels. However, for scores in the 1-2 and 2-3 ranges, the model's performance is suboptimal, with less than 50% of predictions falling within the same buckets as the ground truth. This observation suggests that our model is more adept at recognizing highly similar sentences as well as extremely dissimilar sentences, but struggles when distinguishing between somewhat similar and somewhat dissimilar relationships. The imbalanced training dataset, with more samples in the 0-1, 3-4, and 4-5 ranges, likely contributes to this discrepancy. Addressing this imbalance by incorporating

Sentence 1	Sentence 2	Specific Reason
"How would Hillary Clinton be as a president?"	"What would Hillary Clinton do as the president?"	Topic Consistency but Different Focus
"How do warm and cold fronts form?"	"How does a cold front form?"	Similar Structures with Different Contexts

Table 4: Failure cases for paraphrase detection



(a) Semantic similarity confusion matrix

(b) Distribution of sentence similarity scores

Figure 2: Model performance decomposition for Semantic Textual Similarity

diverse training samples across all score ranges could potentially improve the model's performance in distinguishing similarities and dissimilarities between sentence pairs.

7 Conclusion

In this work, we present a comprehensive exploration of various methods to address the challenges in multi-task learning, focusing on a case study involving the fine-tuning of the pretrained BERT model for three distinct downstream tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. Our evaluation demonstrates the effectiveness of our proposed approaches in improving the collective performance of the multi-task learning objective. Moreover, our comparative analysis of the results obtained from single-task training and multi-task fine-tuning reveals that the latter approach achieves higher scores in individual tasks than the scores obtained from specialized fine-tuning. This finding underscores the effectiveness of multi-task learning in leveraging shared representations and knowledge across related tasks, ultimately leading to improved generalization and performance.

8 Acknowledgement

We express our sincere gratitude to our mentor, Jingwen Wu, for her invaluable guidance and kindness. The authors of this paper have contributed equally to the writing. Regarding the implementation, Xinyan focused on implementing the baseline method and the optimizations related to interleaved training, gradient surgery, gradient clipping and LoRA. Wei concentrated on the implementation of cross encoding, contrastive learning, adversarial regularization, and specialized fine-tuning.

9 Ethical Considerations

We outline two ethical considerations related to our project.

Data Privary in Language Model Training. While there is an increasing trend of governmental restrictions and public attention towards data privacy in the context of training language models, it remains a fundamentally difficult problem to precisely detect whether a third-party model has used private or copyrighted datasets in their training stage. In our project, we have leveraged pretrained weights for the BERT model; however, except for the training data listed in the original BERT paper,

it is beyond our capabilities to ascertain whether this model has been trained with private data. This problem is exacerbated when AI practitioners use pretrained weights from untrustworthy third parties, who may be motivated to use grey-area data to improve the performance of their model, potentially leading to the leaking of user private data.

Abusive Use of Model. The abusive use of the model is another crucial ethical consideration. While our study focuses on utilizing language models to perform sentiment analysis on sentences, which can be useful in applications such as detecting hate speech on social media platforms, the technology can also be intentionally adapted to incorrectly categorize or tolerate certain biases. Due to the difficulty in tracing the training process, the owners of the model can find excuses to evade responsibility for their abusive use of the model. To mitigate this risk, it is imperative to ensure that models deployed for public use or used for public platforms are thoroughly inspected.

References

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). *SEM 2013 shared task: Semantic textual similarity. In Diab, M., Baldwin, T., and Baroni, M., editors, *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.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations.

Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Fernando, S. and Stevenson, M. (2008). A semantic similarity approach to paraphrase detection.

Gao, T., Yao, X., and Chen, D. (2022). Simcse: Simple contrastive learning of sentence embeddings.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Zhao, T. (2020). SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Joyce, J. M. (2011). *Kullback-Leibler Divergence*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lopez-Paz, D. and Ranzato, M. (2022). Gradient episodic memory for continual learning.

Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T. (2019). Transfer learning in natural language processing. In Sarkar, A. and Strube, M., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. (2018). Overcoming catastrophic forgetting with hard attention to the task. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4548–4557. PMLR.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., and Bethard, S., editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. (2020). Gradient surgery for multi-task learning.

Zhang, J., He, T., Sra, S., and Jadbabaie, A. (2020). Why gradient clipping accelerates training: A theoretical justification for adaptivity.