QuarBERT: Optimizing BERT with Multitask Learning and Quartet Ensemble

Stanford CS224N Default Project

Siyi Gu
Department of Computer Science
Stanford University
sgu33@stanford.edu

Zixin Li
Department of Computer Science
Stanford University
zixinl4@stanford.edu

Ericka Liu
Department of Computer Science
Stanford University
yilin23@stanford.edu

Abstract

Transfer learning has reshaped natural language processing (NLP) by pre-training models on extensive text corpora before fine-tuning them on specific tasks. Yet, the challenge of overfitting due to limited task-specific data and the complex nature of pre-trained models hampers their ability to generalize to unseen data, especially in multitask. In this project, we explore the generalization performance of pretrained BERT models in three downstream tasks: sentiment analysis(SST-5), paraphrase detection (QQP), and semantic textual similarity (STS-B). We integrate a multi-task learning framework enhanced by several key methodologies: ensemble models for task-specific performance optimization, Smoothness-Inducing Adversarial Regularization (AdvReg) for maintaining generalization, Momentum Bregman Proximal Point Optimization for controlled model updates, the unsupervised SimCSE framework for effective contrastive learning in sentence embedding generation, and Low Rank Adaptation for efficient fine-tuning. Our results show that our approach improves model generalization and robustness, leading to strong performance improvements across all three tasks simultaneously, achieving the 7th place on testset leaderboard.

1 Key Information to include

Special thanks to our mentor: Heidi Zhang. We do not have external collaborators and we are not sharing projects. Contribution: We equally contributed to the writing. Siyi: Focused on implementing baseline, Cosine, ensemble and LoRA. Zixin: Focused on implementing the Adversarial Regularization and MBPP. Ericka: Focused on implementing the Adversarial Regularization and Contrastive Learning.

2 Introduction

In the realm of natural language processing (NLP), the advent of transfer learning has been a game-changer, fundamentally altering how models are developed and applied. By adopting a two-phase approach—initially pre-training on vast collections of text corpora before fine-tuning for specific tasks—researchers have pushed the boundaries of what NLP models can achieve, as highlighted by Chen et al. (2021). Such models now excel across a diverse aspect of NLP challenges, setting new benchmarks for performance. Despite these advances, deploying these sophisticated models in multitask settings introduces a set of distinct challenges. Among these, the risk of overfitting due to limited task-specific data and the intrinsic complexity of pre-trained models are prominent, limiting their ability to generalize to unseen data. Multi-Task Learning (MTL) presents a promising solution, fostering the idea that knowledge acquired from one task can enhance performance in others when multiple tasks are learned concurrently (Liu et al., 2019). Yet, previous studies (Yu et al., 2020) found that conflicting gradients among different tasks may induce negative interference. Various approaches have been explored to remedy negative interference, such as Adversarial Regularization (Liu et al., 2019), Bregman Gradient Optimization (Liu et al., 2019), and Gradient Surgery (Yu et al., 2020)

In response to these challenges, our study presents an innovative ensemble architecture that leverages both the generalizability of Multi-Task Learning and the specialized knowledge of task-specific models. We initiated our approach by implementing and training BERT on sentiment analysis using the SST and CFIMDB datasets for single-task classification. Building upon this foundation, we adopted the Multi-Task framework from Liu et al. (2019), which uses shared BERT embeddings across tasks while tailoring loss functions to meet specific objectives. To address overfitting and improve generalization, we implemented Adversarial Regularization and Momentum Bregman Proximal Point Optimization techniques from scratch, drawing insights from Jiang et al. (2020). Additionally, our research introduces an novel "pretraining" phase with Contrastive Learning, aiming to improve the model's ability to capture semantic nuances and generalization. Our contributions extend to the creation of an advanced ensemble architecture for multitasking, combining four Multi-Task BERT models, each fine-tuned on specific tasks, with a universally fine-tuned model across all tasks to achieve superior performance. To counter the computational demands of this ensemble approach, we independently implemented the Low-Rank Adaptation (LoRA) technique, achieving a balance between computational efficiency and robust model performance.

3 Related Work

With the recent advancements in deep learning and transformer architectures, pre-trained models have revolutionized the field of natural language processing. The introduction of BERT by Devlin et al. (2019) utilizes deep transformer models that are pretrained on large corpora for downstream tasks. BERT marked a significant leap in leveraging deep learning for language understanding. This foundation has inspired a wide range of research directions, including Multi-Task framework (Liu et al., 2019) and Contrastive Learning (Gao et al., 2022).

While BERT set the foundation for advanced language models, Multi-Task learning emerged as a strategy to leverage shared knowledge across different tasks and enhance model generalization (Liu et al., 2019). This approach aligns with our methodology, wherein we extend BERT's capabilities to handle sentiment analysis, paraphrase detection, and semantic textual similarity simultaneously. The ensemble techniques integrate specialized knowledge from task-specific models to enhance overall performance, despite being computationally intensive.

Jiang et al. (2020) propose Adversarial Regularization as a robust technique for combating overfitting, a challenge often faced when fine-tuning pre-trained models for specialized tasks. Adversarial Regularization, combined with Momentum Bregman Proximal Point Optimization drawing from the same study, refine the model's stability and performance while preserving the rich linguistic comprehension that BERT encapsulates.

Inspired by the success of self-supervised learning, Contrastive Learning, especially in its unsupervised form as seen in SimCSE by Gao et al. (2022), encourages models to maximize the similarity between embeddings of semantically similar data points while minimizing the similarity between embeddings of dissimilar data points. This approach not only refines the models' ability to capture subtle semantic differences but also contributes to a more uniform embedding space and enhances the alignment of semantically related pairs.

The multi-head attention mechanisms and extensive parameters of transformer architectures require considerable memory and computational resources, posing challenges for researchers with limited access to advanced hardware. The introduction of Low Rank Adaptation (LoRA) represents a key advancement in addressing these computational challenges (Hu et al., 2021). By strategically inserting low-rank matrices to approximate the changes in the weight matrices of pre-trained models, LoRA presents a novel way to adapt large models more efficiently. This approach not only alleviates the computational and memory intensity but also retains the nuanced understanding learned during pre-training, enabling more sustainable model development.

4 Approach

Our approach aims to enhance BERT's performance in multitask learning, specifically its ability to simultaneously perform the following three tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. For the baseline, we implemented and trained the BERT model for sentiment analysis on the SST and CFIMDB dataset (See Table 1), leveraging its powerful language

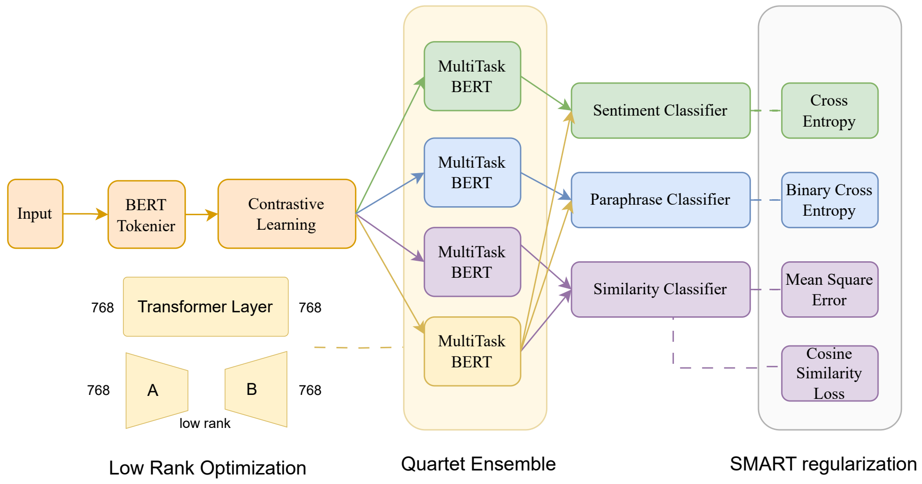

Figure 1: Overview of Multitask Learning Framework and Quartet Ensemble

understanding capabilities to perform sentiment classification (Devlin et al., 2019). Secondly, we further integrate Multi-Task framework to enhance its generalization ability and sets our baseline for multi-task learning based on pre-trained BERT. We designed Multi-Task BERT following Liu et al. (2019), where the BERT embeddings are shared across different task-specific layers. For tasks such as paraphrase detection and semantic textual similarity, we utilize the BERT model to process pairs of sentences, producing embeddings for each. Depending on the task, BERT embeddings are then fed into distinct linear layers with dropout to handle three tasks with varied objectives. Specifically, we use cross entropy loss for sentiment analysis, binary cross entropy loss for paraphrase detection, and Mean-Squared Error for semantic textual similarity. We explored two variants of the baseline, one where the pretrained BERT parameters frozen and another that fine-tunes the BERT parameters, allowing us to assess the effect of updating BERT's parameters alongside task-specific layers.

As shown in Fig. 1, we further explore several extensions and designed a novel ensemble architecture for multi-task learning. Our framework is first fine-tuned through a contrastive learning stage based on the Quora Question Pairs, which effectively primes the model for identifying nuanced textual similarities. Subsequent to this stage, we designed a quartet ensemble of Multi-Task BERT, where three of them fine-tuned on separate tasks—sentiment analysis, paraphrase detection, and semantic similarity evaluation, and one fine-tuned on all three tasks. Ultimately, the outputs from these classifiers are aggregated, harmonizing the generalization benefits of Multi-Task BERT with the specialized knowledge from task-specific models. Since the ensemble model incurs higher computational demands, we aim to further incorporate LoRA, an efficient parameterization technique, into our ensemble architecture, therefore alleviating the computational constraints.

4.1 Ensemble

Besides our implmentation of Multi-Task BERT with task-specific layers and shared BERT layers, we further explored the performance of ensemble models. Specifically, we trained three BERT models each focused on its specfic tasks, maximizing its performance on the individual task independently. We further combined our Multi-Task BERT with the new ensemble models to make the final predictions. While such ensemble model is computationally expensive, we hope to leverage both the generalizability of Multi-Task BERT and the specialized knowledge of task-specific models. For classification tasks, we take average of the probability to obtain the predicted class. For regression tasks, we simply take average of the prediction scores to get the final output.

4.2 Contrastive Learning

We implemented the unsupervised SimCSE (Simple Contrastive Sentence Embedding) framework from scratch to fine-tune our pre-trained model (Gao et al., 2022). SimCSE is predicated on the idea that a sentence, when passed through the model multiple times with non-deterministic dropout, yields varied but semantically consistent embeddings. The training objective for SimCSE is formalized as minimizing the negative log likelihood of a softmax distribution over similarities of sentence embeddings:

$$\ell_i = -\log \frac{e^{\text{sim}(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(h_i, h_j^+)/\tau}}, \quad \text{sim}(h_1, h_2) = \frac{h_1^\top h_2}{\|h_1\| \cdot \|h_2\|} \tag{1}$$

where h_i and h_i^+ are the embeddings of a sentence and its dropout-augmented counterpart, N is the batch size, and τ is the temperature hyperparameter that scales the similarity scores.

In our implementation, we employ an unsupervised learning method by utilizing dropout to create positive pairs (x, x^+) for all of our datasets where x^+ is a sentence semantically related to x. In the context of the unsupervised SimCSE framework shown in Fig. 2, x^+ is simply another instance of x passed through the model with different dropout masks. We denote the embeddings of x and x^+ by h and h^+ respectively, obtained by applying the BERT encoding function f such that $h = f(x)$ and $h^+ = f(x^+)$. Despite x and x^+ being the same in terms of content, the resultant embeddings h and h^+ differ due to the stochastic nature of dropout, thereby serving as effective data augmentation. Negative pairs are taken as all other combinations of sentences in a training batch. This strategy ensures a clear distinction between positive and negative examples, essential for the contrastive learning process to effectively guide the model in developing a rich and nuanced understanding of sentence embeddings.

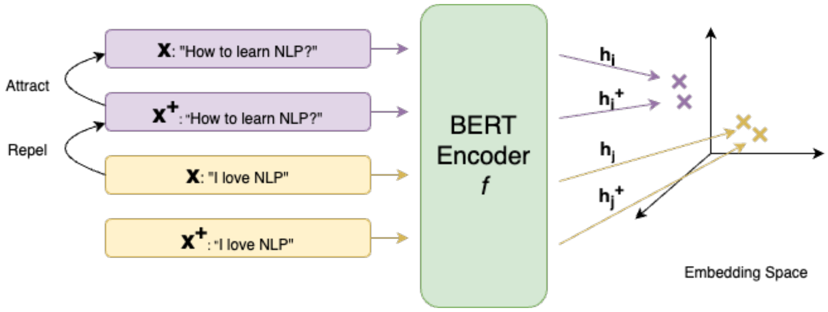

Figure 2: Unsupervised SimCSE Illustration

4.3 Adversarial Regularization

BERT is trained on a comprehensive and rich corpus, providing it with a broad understanding of language. Therefore, when fine-tuning BERT for specific downstream tasks, it's crucial to avoid excessively altering the model's weights. We introduce **Adversarial Regularization** by adding random small noise during training to prevent overfitting on single task and enforcing smoothness to make the model's embeddings more robust to perturbations. This procedure ensures that minor perturbations to the input do not lead to disproportionately large changes in the output, a property that is essential for the model to generalize well to new data. Our implementation of this technique follows the principles outlined by Jiang et al. (2020), i.e.

$$\min_\theta F(\theta) = \mathcal{L}(\theta) + \lambda_s R_s(\theta), \tag{2}$$

The task-specific loss $\mathcal{L}(\theta)$ and the adversarial regularizer $R_s(\theta)$ can be further defined as $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$ and $R_s(\theta) = \frac{1}{n}\sum_{i=1}^{n} \max_{\|\tilde{x}_i - x_i\|_p \le \epsilon} \ell_s(f(\tilde{x}_i;\theta), f(x_i;\theta))$, where $\ell(\cdot,\cdot)$ is the target-specific loss function, $\epsilon > 0$ is a hyperparameter. For classification tasks, we compute symmetrized KL-divergence $\ell_s(P, Q) = D_{\text{KL}}(P\|Q) + D_{\text{KL}}(Q\|P)$; and for regression tasks, we compute mean square error $\ell_s(p, q) = (p - q)^2$.

We then combine the adversarial loss $R_s(\theta)$ with the task-specific loss $\mathcal{L}(\theta)$, balanced by a hyperparameter $\lambda_s > 0$. By integrating the adversarial loss $R_s(\theta)$ with the task-specific loss $\mathcal{L}(\theta)$, and modulating their relative influence through the hyperparameter λ_s, Adversarial regularization ensures that a model's output predictions' smoothness under minor input perturbations. This is particularly helpful in low-resource domain task because the model will be less likely to overfit the small amount of data it has been trained on and more likely to perform well on unseen data.

4.4 Momentum Bregman Proximal Point Optimization

To mitigate overly aggressive updates and to further prevent overfitting, we introduces the application of a **Bregman Proximal Point Optimization** (BPPO) method, inspired by Jiang et al. (2020). This approach significantly enhances our model's generalization capability across various tasks through a Bregman divergence-based regularization framework. Specifically, at iteration $t+1$, model parameters θ_{t+1} are updated using:

$$\theta_{t+1} = \text{argmin}_{\theta} \left(F(\theta) + \mu D_{\text{Breg}}(\theta, \theta_t) \right), \tag{3}$$

where $\mu > 0$ serves as a regulatory tuning parameter, and $D_{\text{Breg}}(\theta, \theta_t) = \frac{1}{n} \sum_{i=1}^{n} \ell_s(f(x_i; \theta), f(x_i; \theta_t))$, with ℓ_s being the symmetric KL divergence or mean square error, depending on the task. This divergence measures discrepancies in model outputs between iterations, ensuring consistency with previous learning and preserving out-of-domain insights inherent in the pre-trained model. Additionally, we augmented the Bregman proximal point method with a momentum component, defined as:

$$\theta_{t+1} = \text{argmin}_{\theta} \left(F(\theta) + \mu D_{\text{Breg}}(\theta, \tilde{\theta}_t) \right), \tag{4}$$

in which $\tilde{\theta}_t = (1-\beta)\theta_t + \beta\tilde{\theta}_{t-1}$ calculates a weighted average of last parameters. The momentum parameter $0 < \beta < 1$, fine-tunes the extent of parameter averaging. This mechanism not only speeds up the optimization but also ensures more stable and effective learning outcomes.

4.5 Data Sampling

After evaluating the performance of baseline models, we observed a common issue: they exhibited higher accuracy in paraphrasing tasks while showing low accuracy in textual similarity and sentiment analysis. This discrepancy stems from the imbalance in the dataset sizes used for fine tuning. Such disparities in dataset size lead to models overfitting the larger datasets and underperforming on the smaller ones, thereby restricting their generalization ability. To address this challenge and simultaneously perform three downstream tasks, we introduced a batch-level data samping method. By selecting the minimum number of batches from all training datasets, and updating each models' weights with a single pass for each batch, we ensured balanced exposure to each task. This approach promotes uniform learning across tasks, irrespective of their individual dataset sizes.

4.6 Cosine Similarity

Inspired by SBERT (Reimers and Gurevych, 2019), we adopt the computation of **cosine similarity** between sentence representations for STS-B ($\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$, where $\mathbf{u}$ and $\mathbf{v}$ are sentence embeddings obtained from BERT). In this way, we adjust the model's embeddings to maximize the cosine similarity for semantically similar sentences and minimize it for dissimilar ones. Since the similarity score is in range of (-1,1), we further apply a sigmoid function and scale the output to span 0 to 5 to match true label range in the STS dataset. Since BERT embeddings are semantically-rich, similar sentences should have similar embeddings. Cosine similarity intuitively computes similarity between two vectors, which is highly correlated with semantic similarity. It enables the model to better capture semantic similarity between sentences, which is especially beneficial for tasks like STS, where the goal is to quantify the degree of semantic equivalence between two sentences.

4.7 Low Rank Adaptation

As transformers and ensembles are generally computationally expensive, we integrate Low Rank Adaptation (LoRA) into our framwork, alleviating storage and computation burden required to train large language models for multitask learning (Hu et al., 2021). In Multi-head attention, each head can be computed as

$$\text{head}_i = \text{Attention}(Q\mathbf{W_i}^Q, K\mathbf{W_i}^K, V\mathbf{W_i}^V) \tag{5}$$

where $\mathbf{W_i}^Q, \mathbf{W_i}^K, \mathbf{W_i}^V$, are weight matrices specific to each head for queries Q, keys K, and values V. LoRA can be further applied within the transformer architecture to adapt these weight matrices $\mathbf{W_i}^Q, \mathbf{W_i}^K$ and $\mathbf{W_i}^V$ efficiently. In the LoRA framework, the idea is to adapt these weight matrices in a computationally efficient way. Instead of directly learning the full weight matrices, LoRA introduces two low-rank matrices to be updated $\mathbf{A}$ and $\mathbf{B}$ such that

$$\Delta\mathbf{W} = \mathbf{AB} \tag{6}$$

where $\mathbf{A}$ and $\mathbf{B}$ are much smaller in size compared to $\mathbf{W}$, making the adaptation computationally efficient and significantly reducing the number of trainable parameters. Inspired by BERT-LoRA-TensorRT, we implemented a low-rank adapted linear layer from scratch with cutomized rank of the A and B matrices.

5 Experiments

5.1 Data

We use three datasets for our multitask learning framework, each with a different sentence-level task:

- **Stanford Sentiment Treebank (SST-5)** (Socher et al., 2013) is used for sentiment analysis and contains 11,855 single sentence reviews, where the reviews are categorized into five groups: negative, somewhat negative, neutral, somewhat positive, positive. The dataset is split to 8,544 train, 1,101 dev, and 2,210 test samples.
- For paraphrase detection, we use the **Quora Question Pairs (QQP)** dataset (Fernando and Stevenson, 2008) comprising 400k question pairs with binary labels indicating whether the questions are paraphrases of each other. The dataset is split to 141,506 train, 20,215 dev, and 40,431 test samples.
- **Semantic Textual Similarity(STS-B)** (Agirre et al., 2013) consists of 8,628 sentence pairs, where each pairs of sentences are scored based on their semantic similarity on a 0 to 5 scale. The number of samples in train, dev, and test are 6,041, 864, and 1,726 respectively.

5.2 Evaluation Method

We evaluate the performance on the SST and Quora dev set and test set with accuracy, a measurement of how well predicted labels match the ground truth labels. For the STS dataset, since it's range from 0 to 5, we use Pearson correlation to evaluate the linear relationship between predicted scores and ground-truth.

5.3 Experimental Details

We implement BERT architecture and use pretrained weights from the Hugging Face Transformers library. For the baseline, we test the performance of both the updated (Baseline-finetune) and frozen (Baseline-pretrain) BERT parameters. For the extensions, we consistently update the BERT parameters. By default, we run 10 epochs with a learning rate of 1e-5 for fine-tuning and 1e-3 for pretrain. The dropout probability is set to 0.1 and batch size is set to 16. Due to memory constraints, we set the batch size to 8 when fine-tuning with SMART loss. For the adversarial regularization, we set $\lambda = 5$ and $\epsilon = 1e - 5$. For Bregman optimization, we set $\sigma = 1e - 5$, $\beta = 0.995$, and $\mu = 1$. For ensemble models, we take average of each of the sub-models' predictions to get our final predictions. Additionally, we set the contrastive learning weighting parameter, λ_contrastive = 0.1 to balance the contrastive learning objective in our multitask learning framework.

Table 1: Quantitative Result on Part 1. Accuracy comparison of BERT on SST and CFIMDB.

	Ours		**Reference**	
	SST	CFIMDB	SST	CFIMDB
dev-pretrain	0.417	0.784	0.390	0.780
dev-finetune	0.537	0.959	0.515	0.966

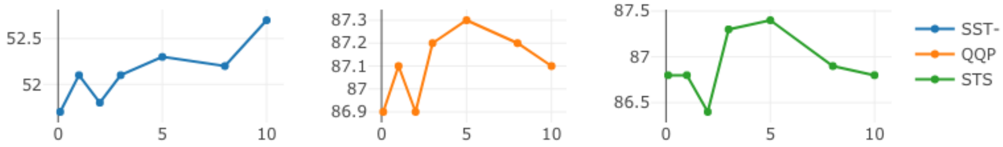

Figure 3: Model Performance with Varied λ

Table 2: Performance Comparison of BERT with different extensions on SST-5, Quora, and STS-B Tasks. The table presents accuracy metrics for SST-5 and Quora, and Pearson correlation coefficients for STS-B, with higher values indicating better model performance.

Model	Parameters	**SST-5** Acc ↑	**QQP** Acc ↑	**STS-B** Corr. ↑	Avg
Baseline-pretrain	2M	0.318	0.432	0.228	-
Baseline-finetune	112M	0.411	0.543	0.348	-
sampling	111M	0.513	0.816	0.753	-
SMART-sampling($\lambda = 5, \epsilon = 1e - 5$)	112M	0.523	0.868	0.832	-
SMART-sampling-cosine	112M	0.518	0.871	0.790	-
SMART-sampling-SimCSE	112M	0.529	0.878	0.869	-
SMART-sampling-SimCSE-Ensemble	449M	**0.537**	**0.874**	**0.869**	**0.782**
SMART-sampling-SimCSE-Ensemble-LoRA	12M	0.490	0.865	0.859	-
Best-test performance (Ranked 7)		**0.547**	**0.875**	**0.875**	**0.787**

Table 3: Model Performance with Varied Learning Rate and Hidden Dropout Probability.

Learning Rate	SST	QQP	STS
1e-5	**0.521**	**0.871**	**0.875**
2e-5	0.505	0.867	0.868
3e-5	0.490	0.867	0.854
5e-5	0.431	0.836	0.854

Dropout Rate	SST	QQP	STS
0.1	**0.529**	**0.878**	**0.869**
0.2	0.526	0.875	0.866
0.3	0.519	0.876	0.867

5.4 Results

Tab. 1 and Tab. 2 present our results on sentiment analysis and multi-tasking respectively. Upon integrating several extensions, as shown in Tab. 2, our model exhibits significant performance enhancements across all three tasks, with best performance on paraphrasing detection. For the dev set, we achieved an accuracy of 0.537 on SST, 0.874 on QQP, and pearson correlation of 0.869 on STS-B, leading to a overall performance score of 0.782. For the test set, we achieved an accuracy of 0.547 on SST, 0.875 on QQP, and pearson correlation of 0.875 on STS-B, leading to a overall performance score of 0.787 and ranked 7 on the leaderboard. Results showed that all extensions improved the model's performance and combining them we outperform the baseline by a notable margin. As expected, since the purpose of LoRA is to enable faster training and reduce memory usage, it in fact leads to worse performance on all three tasks. Yet the usage of LoRA enables us to reduces the trainable parameters to approximately 2.6%, showcasing its effectiveness in reducing computational costs.

We further provide results with hyperparameter tuning on learning rate, dropout probability and λ in adversarial regularization, as shown in Table 3 and Fig. 3. We find that during fine-tuning, model performs the best with a setting of learning rate of 1e-5, dropout probability of 0.1 across all three tasks. When tuning λ, we find out a lower value of λ leads to better performance on SST yet QQP and STS-B have best performance around $\lambda = 5$.

6 Analysis

We further visualize the distribution of sentiment scores and similarity scores in Fig. 4 with our current best model. Fig. 4 (left) shows that our model is good at capturing the overall score distribution but suffer when distinguish sentiments with slight differences(neutral and somewhat positive). Fig. 4 (right) shows that our model is correctly predicting the overall distribution for the STS task but is too conservative when predicting extreme similarity scores.

In the SST-5 development dataset, the majority (99%) of errors occurred when the model's predictions deviated by only one or two levels from the actual labels. This pattern underscores the model's difficulty in accurately discerning sentiment intensity, especially in distinguishing between "positive" (3), "very positive" (4), "negative" (1), and "very negative" (0) sentiments. For instance, the sentence: "Not only is Undercover Brother as funny, if not more so, than both Austin Powers films, but it's also one of the smarter, savvier spoofs to come along in some time," received an actual sentiment of 4 (Very Positive) but was predicted as 3 (Positive) by our model. This discrepancy indicates a challenge in precisely predicting sentiment levels. Moreover, the model appears to struggle with detecting sentiment in sentences predominantly composed of neutral wording. A case in point is "The Iditarod lasts for days - this just felt like it did," which offers a critique of the movie's pacing by negatively comparing its length to the prolonged Iditarod race, receiving an actual sentiment of 0 (Very Negative) yet predicted as 2 (Neutral). The model's inability to discern the critique's negative sentiment points to a potential improvements if we can enhanced sentiment detection within neutral contexts. Additionally, certain labels within the dataset may be deemed questionable or unreasonable. For example, "Reign of Fire looks as if it was made without much thought – and is best watched that way." was categorized as 3 (Positive) but anticipated as 1 (Negative) by the model, suggesting potential inconsistencies in label assignment.

In the QQP dev dataset, our model's performance appears compromised by learning lexical similarity instead of semantic content. It fails to recognize paraphrases for the sentence pair "What are the best ways to learn English?" and "How can I improve my English skills?". This indicates a reliance on word matching rather than understanding underlying meanings. Another example is the model mistook "How do I register for Star Alliance?" and "What benefits do Star Alliance members get?" for paraphrases. Moreover, it also struggles with recognizing the general equivalence of specific instances, for example, not seeing the similarity between "What is the implication of free education in rte?" and "What is the implication of free education in the right to education act?".

In the STS-B development dataset, our model appears to have difficulty accurately discerning the varied meanings of words. For instance, it gives a similarity score of 3.26 to the sentences "Work into it slowly." and "It seems to work.", despite their completely different meaning.

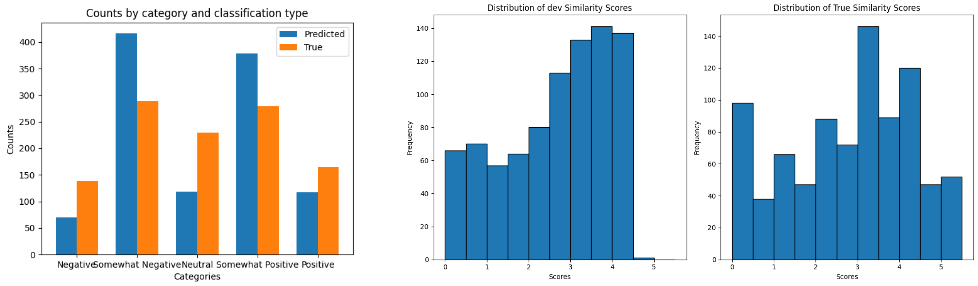

Figure 4: (Left) Distribution of predicted sentiment labels and true sentiment labels for the SST dataset. (Right) Comparison of distribution of predicted similarity scores (left) and actual similarity scores (right) for the STS task. Scores range from 0 (unrelated) to 5 (equivalent meaning).

7 Conclusion

In conclusion, our project illustrates that a multitask learning framework, enhanced with Adversarial Regularization, Momentum Bregman Proximal Point Optimization, and other innovations like ensemble models and SimCSE contrastive learning significantly boosts BERT's generalization across diverse NLP tasks. Our experiments across three different datasets have yielded promising results, notably achieving the 7th place on the test set leaderboard. This underscores the success of our methods and the promise of these combined techniques in tackling overfitting and improving model generalization within NLP.

Despite these achievements, our work is not without limitations. The computational expense of ensemble models and the balance between model complexity and performance remain areas for further exploration. Moreover, the relatively conservative predictions in semantic textual similarity tasks suggest room for refinement in capturing extreme semantic variances.

As we look to the future, there are several areas of improvement. Exploring various regularization and contrastive learning methods, fine-tuning strategies, and applying our approach to more NLP tasks and languages are all promising paths. With continued research and development, we are optimistic about the potential to further elevate the capabilities of NLP models in understanding and processing human language.

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.

Shijie Chen, Yu Zhang, and Qiang Yang. 2021. Multi-task learning in natural language processing: An overview. *arXiv preprint arXiv:2109.09138*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.