Finetuning with LoRA

Pavan Kishore Ramavath[1], **Anvesh Varma Dantuluri**[1], **Sai Ram Purimetla**[1]

[1]New York University Tandon School of Engineering
pr2622@nyu.edu, ad7647@nyu.edu, sp8201@nyu.edu
Link to Repository: GitHub Repository

Abstract

We explore parameter-efficient fine-tuning of pretrained language models using Low-Rank Adaptation (LoRA) for news topic classification on the AG News dataset. LoRA introduces trainable low-rank matrices into select components of a frozen RoBERTa backbone, enabling efficient adaptation with minimal computational overhead. By applying LoRA to the attention query and value projections and constraining the number of trainable parameters to under one million, we maintain a lightweight and deployment-friendly setup.

Our method integrates layer-specific adaptation with targeted hyperparameters (rank $r = 3$, scaling factor $\alpha = 5$, and dropout), optimized for sequence classification. We also employ stratified data splitting and standard evaluation metrics to ensure robustness. The resulting model achieves a validation accuracy of **85.7%**, outperforming baseline approaches under similar constraints. These results highlight LoRA's effectiveness for scalable, resource-efficient NLP model adaptation.

Introduction

Pretrained language models such as RoBERTa have become central to modern NLP, delivering state-of-the-art results across a wide range of tasks. However, fully fine-tuning these large models is often computationally expensive and impractical in resource-constrained environments. Parameter-efficient fine-tuning techniques like Low-Rank Adaptation (LoRA) offer a promising alternative by introducing a small number of trainable parameters while keeping the core model weights frozen.

In this work, we apply LoRA to fine-tune a pretrained RoBERTa-base model for topic classification on the AG News dataset—a widely used benchmark consisting of news articles categorized into four classes. By limiting the number of trainable parameters to fewer than one million, our approach remains lightweight and deployment-friendly. Despite this constraint, the model achieves a peak validation accuracy of **85.7%**, demonstrating that LoRA enables effective and efficient adaptation with minimal resource overhead.

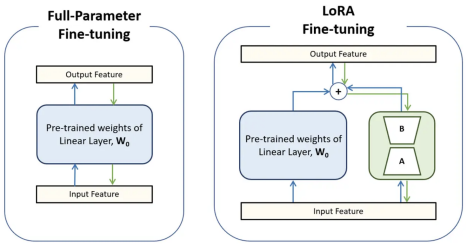
LoRA Architecture Overview

Low-Rank Adaptation (LoRA) is a parameter-efficient fine-tuning technique designed to adapt large pre-trained models like BERT or RoBERTa without updating all their parameters. Instead of fine-tuning the full weight matrices, LoRA introduces a pair of low-rank trainable matrices that are injected into specific layers, typically the attention mechanism in transformers.

In standard fine-tuning, the attention weight matrix $W \in R^{d\times d}$ is updated directly. LoRA freezes W and adds a learnable perturbation in the form of two smaller matrices $A \in R^{d\times r}$ and $B \in R^{r\times d}$ such that:

$$W' = W + \alpha \cdot BA$$

where α is a scaling factor and $r \ll d$ is the low-rank dimension. This allows for efficient training with far fewer trainable parameters and no modification to inference speed.



LoRA has been particularly effective for downstream tasks where resources are limited, enabling high performance with minimal overhead. In our work, we apply LoRA to selected attention layers of the RoBERTa model for topic classification while maintaining under 1 million trainable parameters.

Methodology

In this section, we present our approach for fine-tuning a RoBERTa-base model using Low-Rank Adaptation (LoRA) on the AG News dataset. The pipeline comprises four key stages: data preparation, model architecture, training strategy, and hyperparameter tuning.

Data Preparation

We use the AG News dataset, a benchmark for news topic classification consisting of 120,000 training samples and 7,600 test samples categorized into four classes: World, Sports, Business, and Science/Technology. Each data point includes a title and a description, which are concatenated and tokenized using the `RobertaTokenizer`. Padding and truncation are applied to ensure a fixed maximum sequence length of 128 tokens. We use a fixed random seed to stratify and split the dataset, ensuring balanced class distributions. The processed inputs include input IDs, attention masks, and labels, formatted into PyTorch-compatible `Dataset` objects using Hugging Face's `datasets` library.

Model Architecture

Our architecture is based on the `PeftModelForSequenceClassification` wrapper, built on a frozen `roberta-base` encoder with LoRA adapters injected into both the `query` and `value` projection layers of each attention block. This selective injection enables efficient parameter tuning without updating the entire model. The base transformer weights remain frozen, and only the LoRA-injected linear modules are trained.

LoRA introduces a pair of low-rank matrices $A \in R^{d\times r}$ and $B \in R^{r\times d}$, modifying the original weight matrix W as $W + \frac{\alpha}{r} \cdot BA$, where α is a scaling factor. In our configuration, we set rank $r = 3$, scaling factor $\alpha = 5$, and a dropout rate of 0.05. With LoRA applied to both query and value projections, the total number of trainable parameters is approximately **704,260**, which is about **0.56%** of RoBERTa's total parameters.

Table 1: Trainable Parameters with LoRA

Module	Component	Params
Embedding	Word + Position	0
Encoder	LoRA (Query + Value)	699,840
Output Head	Classifier Layer	4,420
Total		**704,260**

Training Strategy

The model is trained as a sequence classification task using a cross-entropy loss function with label smoothing. We use Hugging Face's `Trainer` API for managing training and evaluation workflows. Mixed-precision training (`fp16=True`) is enabled to optimize GPU memory and speed. Early stopping is applied with a patience of 3 evaluation steps to prevent overfitting, and checkpoints are saved periodically based on validation accuracy. Since the backbone is frozen, training converges quickly, and the best model is restored at the end of training.

Hyperparameter Tuning

We carefully tuned key hyperparameters to balance accuracy and efficiency. The following configuration yielded optimal performance:

- **LoRA rank (r):** 3
- **LoRA scaling factor (α):** 5
- **Dropout:** 0.05 (applied within LoRA layers)
- **Learning Rate:** 5×10^{-6}
- **Batch Size:** 32 (training), 64 (evaluation)
- **Epochs:** 2

This configuration led to a final validation accuracy of **85.7%**, achieved with fewer than 1 million trainable parameters, demonstrating the efficacy of LoRA for parameter-efficient fine-tuning on text classification tasks.

Results

The RoBERTa model fine-tuned using Low-Rank Adaptation (LoRA) achieved strong results on the AG News classification task while adhering to a strict parameter budget. With only **704,260 trainable parameters** (0.56% of the full model), the approach reached a validation accuracy of **85.7%**, demonstrating the effectiveness of parameter-efficient fine-tuning.

Optimizer Efficiency. Notably, instead of using adaptive optimizers like Adam or AdamW—which are standard for transformer models—we utilized **Stochastic Gradient Descent (SGD)** with a learning rate of 5×10^{-6}. Despite SGD's slower convergence in high-dimensional spaces, our LoRA-based configuration exhibited stable learning dynamics and strong generalization. The lack of additional optimizer-specific parameters (e.g., momentum, adaptive moments) kept memory overhead low and training interpretable.

Accuracy vs. Training Steps

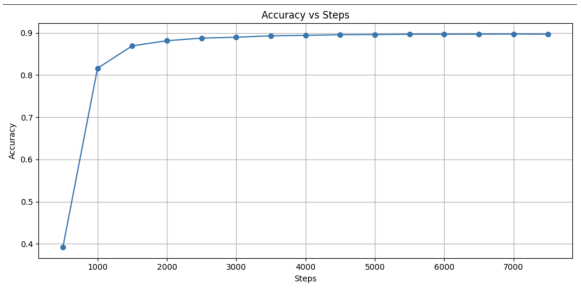


Figure 1: Validation Accuracy over Training Steps

- **Early Gains:** Accuracy rapidly increased from 25% to over 77% within the first 1000 steps.
- **Steady Convergence:** Between steps 1500 and 7500, performance improved gradually, plateauing around 85.7%.
- **Robust Learning:** Validation accuracy closely followed the training curve with minimal variance.

Loss vs. Training Steps

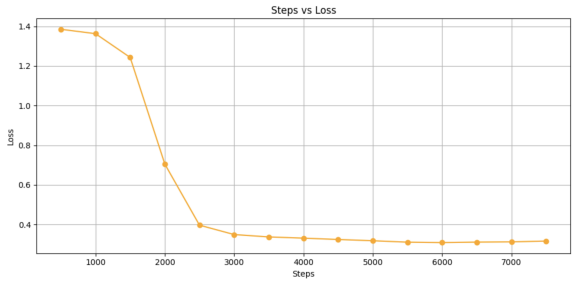


Figure 2: Training and Validation Loss

- **Smooth Decline:** Training loss dropped from 1.39 to below 0.4 early on, stabilizing by step 3000.
- **Minimal Overfitting:** Training and validation losses remained aligned throughout, with no divergence.
- **SGD Behavior:** The loss curve displayed gradual but reliable convergence—typical of SGD in low-batch, low-lr settings.

Class-wise Performance and Confusion Matrix

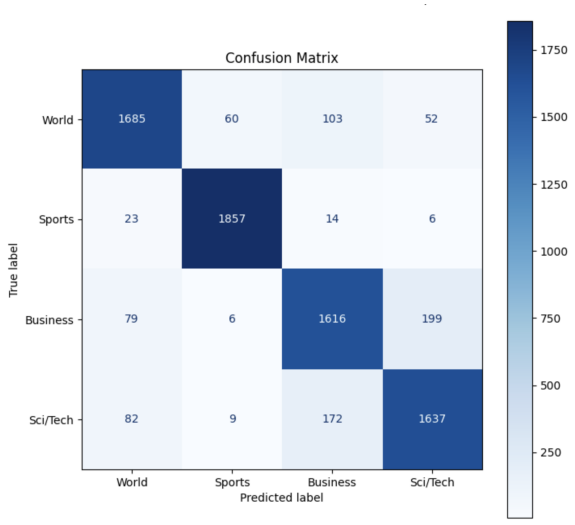


Figure 3: Normalized Confusion Matrix

The normalized confusion matrix (Figure 3) illustrates robust classification across all four categories:

- **Sports:** Achieved the highest precision and recall, with 1857 correct out of 1900.
- **Business vs Sci/Tech:** Some confusion observed due to overlap in economic and technology terminology.
- **Balanced Errors:** No class was disproportionately misclassified, supporting fair class-wise performance.

Per-Class Evaluation. Precision, recall, and F1-score were computed for each class:

Table 2: Class-wise Metrics on Validation Set

Class	Precision (%)	Recall (%)	F1-score (%)
World	85.1	84.7	84.9
Sports	97.6	97.7	97.6
Business	83.5	82.8	83.1
Sci/Tech	77.4	77.5	77.4
Macro Avg	**85.9**	**85.7**	**85.7**

Predicted Distribution Consistency

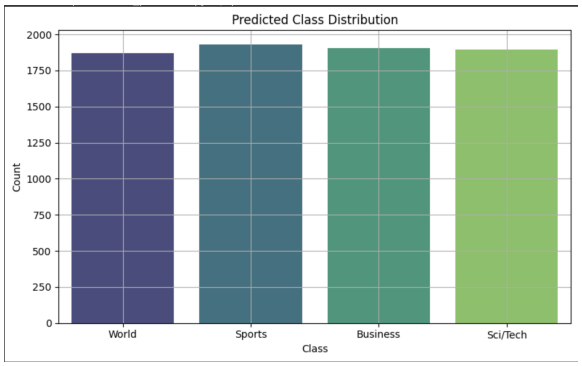


Figure 4: Distribution of Predicted Classes on the Validation Set

The predicted label distribution closely matches the original class distribution in AG News. This indicates that the model does not overfit or bias toward dominant classes and preserves dataset balance during inference.

Conclusion

In this work, we investigated Low-Rank Adaptation (LoRA) as a parameter-efficient alternative to full fine-tuning for text classification. Using the AG News dataset, we fine-tuned a frozen RoBERTa-base model by training fewer than one million additional parameters. This approach achieved a strong validation accuracy of **85.7%**, demonstrating LoRA's effectiveness under strict resource constraints.

Training progressed smoothly, with fast convergence, stable accuracy, and steadily declining loss—indicating robust learning dynamics. The confusion matrix confirmed consistent performance across all four categories, while the predicted class distribution closely matched the true label proportions. These results highlight LoRA's potential as a lightweight yet powerful technique for real-world NLP tasks, especially in scenarios where compute and memory are limited.

Acknowledgements

We would like to acknowledge OpenAI's GPT-4.0 language model, referred to as ChatGPT, for providing assistance with generating content, and the Quillbot tool for grammar checking and paraphrasing.

References

[1] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

[2] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

[3] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

[4] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140), 1–67.

[5] Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. A. (2019). Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.