Parameter-Efficient Fine-Tuning for News Topic Classification using RoBERTa and LoRA

Ali Aslanbayli, Farid Taghiyev

New York University
Tandon School of Engineering

Abstract

This report presents a deep learning approach to fine-tuning a RoBERTa-based language model for news topic classification using Low-Rank Adaptation (LoRA). Leveraging Hugging Face's `transformers` and `peft` libraries, we developed a resource-efficient training pipeline that enables powerful model adaptation with significantly fewer trainable parameters. The model was trained on a labeled news dataset and applied to infer topics on an unlabeled test set. Through visualizations and performance metrics, we demonstrate the practicality and strength of LoRA in production-style NLP systems.

Introduction

Text classification remains a central task in natural language processing, with applications ranging from spam detection to news categorization and sentiment analysis. While large language models like RoBERTa offer exceptional performance, fine-tuning them is often prohibitively expensive in low-resource or rapid prototyping environments.

This project addresses this issue by incorporating Low-Rank Adaptation (LoRA), a parameter-efficient fine-tuning technique. The core idea is to adapt only a small set of low-rank parameters instead of updating the entire transformer network. This enables practitioners to harness state-of-the-art models with dramatically reduced compute requirements. Here is a schematic of how LoRA looks like in Fig. 1.

Our goal was to build an efficient yet effective pipeline for classifying news headlines into predefined categories. Beyond exploring fine-tuning performance, we emphasize deployment feasibility by saving predictions and visualizing inference outputs for model interpretability.

GitHub Repo

The source code alongside all of the output files can be found by visiting https://github.com/aslanbayli/finetuned-roberta

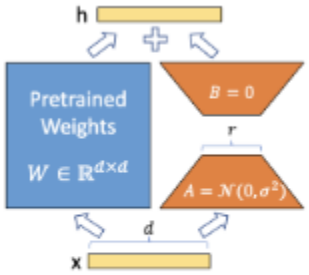

Figure 1: Our reparametrization. We only train A and B.

Figure 1: Block diagram of LoRA

Methodology

Environment Setup

We installed all required libraries such as `transformers`, `datasets`, `evaluate`, `accelerate`, `peft`, and `bitsandbytes`. We also confirmed CUDA availability to ensure training on GPU-accelerated hardware for efficiency.

Dataset Description

The labeled dataset consisted of news headlines, each tagged with one of several topics (e.g., Sci/Tech, Sports, Business, World). It was processed using the Hugging Face `datasets` library, with label fields mapped to class indices via `ClassLabel`. The input text was tokenized using the `RobertaTokenizer` with truncation and padding enabled.

LoRA Model Architecture

Our base model was `RobertaForSequenceClassification`. To enable efficient fine-tuning, we integrated LoRA by injecting trainable low-rank matrices into the attention layers using the following configuration:

- LoRA rank $r = 7$
- Scaling factor $\alpha = 32$
- Applied to query, key, and value projections in attention layers

This reduced the number of trainable parameters to a small fraction of the original model, significantly cutting training time and memory usage.

Training Setup

Model training was conducted using Hugging Face's `Trainer` API. Key training parameters were:

- Learning rate: 2.5×10^{-4}
- Batch size: 32
- Epochs: 2
- Weight decay: 0.01

To prevent overfitting and optimize for generalization, we implemented early stopping based on macro F1 score. Metrics were computed using the `evaluate` library during training.

Model Evaluation

Post-training, the model was evaluated on the validation set achieving an accuracy of 91%. The results confirmed the model's robustness in identifying the correct categories for unseen headlines.

Results

Training Performance

The training curves in Fig. 6, shown below, depict stable convergence over the course of training. As we can see the loss continued so steadily decrease as the training progressed. What is interesting is that all of the metrics including accuracy, precision, recall, and F1-score were nearly identical throughout the training process which indicates that the model is very balanced capable of equally well classifying all classes.

Model Inference and Prediction

After training, the model was used to infer labels on an unlabeled test dataset. Predictions were saved to a CSV file for downstream use, representing successful model deployment.

Additionally, we developed a visualization function that shows the model's confidence distribution across classes for a given input shown in Figures 2, 3, 4,and 5.

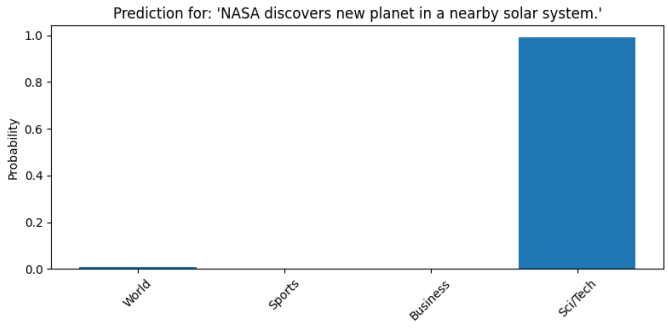

Figure 2: Prediction confidence for: "NASA discovers new planet in a nearby solar system."

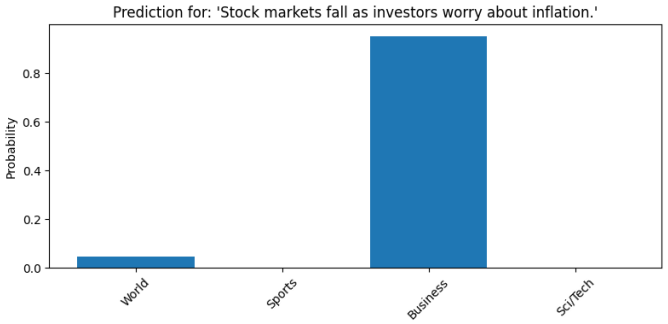

Figure 3: Prediction confidence for: "Stock markets fall as investors worry about inflation."

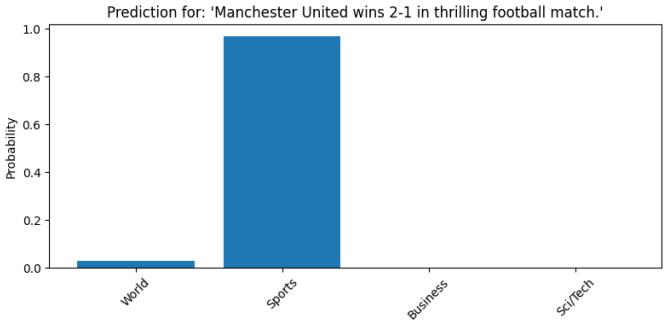

Figure 4: Prediction confidence for: "Manchester United wins 2-1 in thrilling football match."

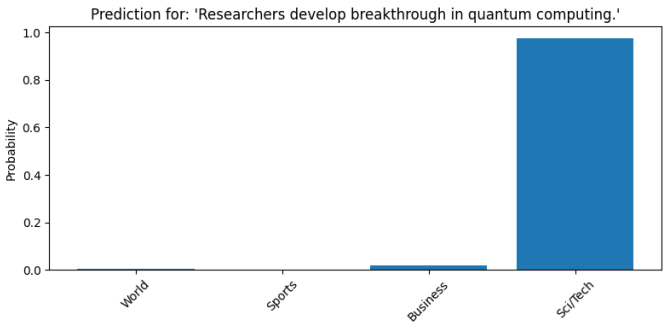

Figure 5: Prediction confidence for: "Researchers develop breakthrough in quantum computing."

Conclusion

This project illustrates the power of parameter-efficient fine-tuning via LoRA for text classification tasks. We successfully trained a RoBERTa-based model on a news dataset and deployed it for batch inference, supported by informative visualizations. The approach dramatically reduces computational cost, proving viable for lightweight applications.

In the future, we plan to:

- Explore prompt tuning or other PEFT methods
- Apply this pipeline to multilingual or noisy text datasets
- Integrate model explainability frameworks like SHAP or LIME

External resources

This report was created with the help of GPT to make the expression of ideas more clear and structured.

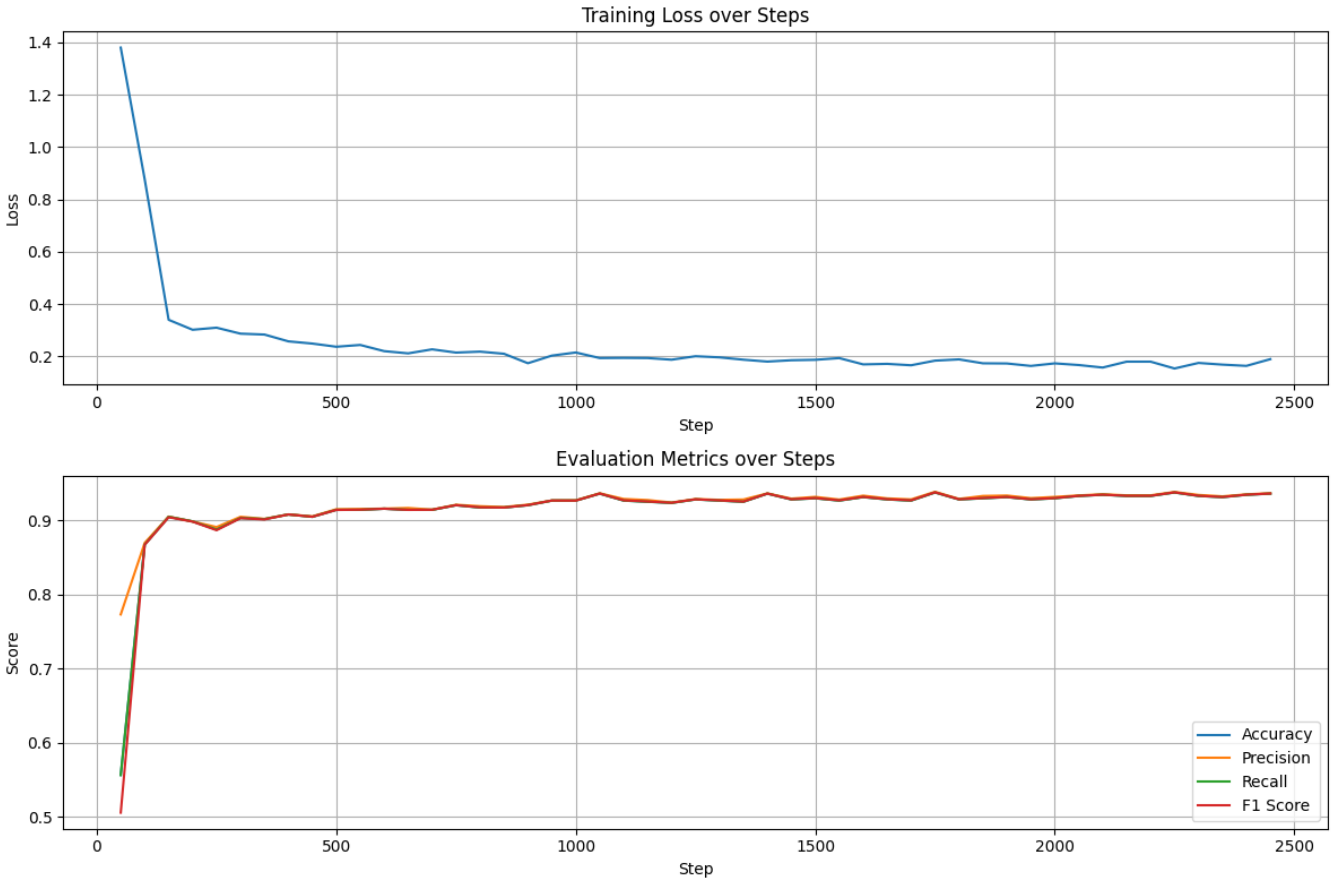

Figure 6: Training loss and metrics for every mini-batch