Efficient Text Classification using Parameter-Efficient RoBERTa Adaptation with LoRA

Chinmay Shringi[*] and **Farnaz Zinnah**[†] and **Mohd Sarfaraz Faiyaz**[‡]

Code & Models (clickable):
https://github.com/fzinnah17/LoRA-finetune-2025

Abstract

This paper presents an optimized approach to text classification using a parameter-efficient adaptation of the RoBERTa model with Low-Rank Adaptation (LoRA). Targeting the AG News dataset classification task, we developed a solution that achieves high accuracy (95.62%) on the evaluation set and a competitive 0.84450 score on the Kaggle leaderboard, while maintaining a strict parameter budget under 1 million trainable parameters. Our methodology combines strategic LoRA configuration optimization, targeted data filtering techniques based on text length distribution analysis, and advanced training strategies including cosine learning rate scheduling with warmup and label smoothing regularization. We systematically evaluated multiple LoRA configurations to determine the optimal rank, alpha values, and target module selection, finding that a comprehensive approach with rank=4, alpha=96, and adaptation of query, key, and value matrices yielded the best results while using only 814,852 trainable parameters (0.65% of the full model). Detailed confusion matrix analysis reveals near-perfect classification for Sports (99%) and Science/Technology (98%) categories, with slightly lower but still impressive performance for World (96%) and Business (89%) categories. Our results demonstrate that careful architectural choices and training optimizations can yield substantial performance improvements without the computational burden of full model fine-tuning, establishing an effective blueprint for parameter-efficient adaptation of large language models in resource-constrained environments.

Introduction

Large pre-trained language models have revolutionized natural language processing, but fine-tuning these models requires substantial computational resources. Parameter-efficient fine-tuning methods, particularly Low-Rank Adaptation (LoRA), offer a promising approach to adapt large pre-trained models while minimizing trainable parameters.

[*]cs7810@nyu.edu
[†]fz675@nyu.edu
[‡]msf9335@nyu.edu

In this work, we apply LoRA to the RoBERTa model for text classification on the AG News dataset (World, Sports, Business, Science/Technology). Our goal is to achieve high classification accuracy while keeping trainable parameters below 1 million. Through optimized LoRA configurations, data filtering, and advanced training methods, we achieved 95.62% evaluation accuracy and a 0.84450 Kaggle score using only 814,852 trainable parameters (0.65% of the total).

LoRA works by inserting low-rank decomposition matrices into the network while keeping pre-trained weights frozen. For weights $W_0 \in \mathbb{R}^{d\times m}$, the adapted weights become

$$W = W_0 + \Delta W = W_0 + BA,$$

where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times m}$, and $r \ll \min(d, m)$. This approach dramatically reduces trainable parameters while maintaining model expressiveness, making it ideal for resource-constrained environments.

Methodology

Our approach follows a systematic framework for parameter-efficient model adaptation, focusing on four key areas: LoRA configuration optimization, data preparation and filtering, training optimization, and comprehensive evaluation.

LoRA Architecture Details

When implementing LoRA for the RoBERTa model, we focus on the self-attention mechanism in each Transformer layer, which comprises query (Q), key (K), and value (V) projections of shape $(\text{hidden\_size}\times\text{hidden\_size})$. In the standard Transformer:

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V.$$

With LoRA, each weight matrix $W \in \mathbb{R}^{d\times m}$ (i.e. W_Q, W_K, W_V) is adapted as

$$W = W_0 + \Delta W = W_0 + BA, \quad \Delta W = \frac{\alpha}{r}\,BA,$$

where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times m}$, and $r \ll \min(d, m)$. We initialize $A \sim \mathcal{N}(0, 1/\sqrt{r})$ and $B = 0$ so that $\Delta W = 0$ at start.

For our Comprehensive configuration ($r = 4, \alpha = 96$) targeting all three projections:

Per head: $768 \times 4 + 4 \times 768 = 6{,}144$,

Per layer: $3 \times 6{,}144 = 18{,}432$,

All 12 layers: $12 \times 18{,}432 = 221{,}184$,

Classification head: $593{,}668$,

Total trainable: $221{,}184 + 593{,}668 = 814{,}852$ (0.65% of model).

LoRA Configuration Optimization

We systematically explored combinations of:

1. **Rank** (r): 2, 3, 4.
2. **Alpha** (α): 16,...,128.
3. **Target modules**: from minimal (query only) to comprehensive (query, key, value).

   Configuration 1 (Minimal): r=2, α=16, targets=[query]
   – Trainable parameters: 188,928

   Configuration 2 (Balanced): r=3, α=32, targets=[query,value]
   – Trainable parameters: 566,784

   Configuration 3 (Comprehensive): r=4, α=96, targets=[query,key,value]
   – Trainable parameters: 814,852

   Configuration 4 (Focused Strong): r=2, α=128, targets=[query,key,value]
   – Trainable parameters: 407,426

Data Analysis and Filtering

We analyzed the AG News training set's text-length distribution (median 37 words, 95th percentile 53), shown in Figure 1. We filtered out examples with length < 20 or > 70 words (middle 80%) and held out 640 balanced examples for validation. This focused the model on high-quality, representative inputs and improved generalization.

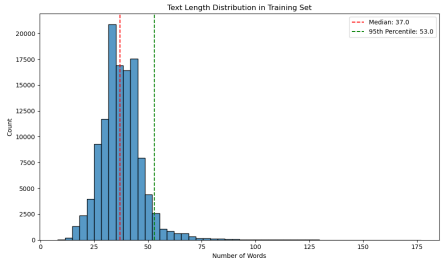

Figure 1: Text-length distribution in AG News (pre-filter).

Training Optimization

We developed a comprehensive training pipeline with numerous optimizations:

- **Optimizer selection:** AdamW with weight decay of 0.005, which decouples weight decay from the learning rate schedule for better generalization.
- **Learning rate scheduling:** Cosine learning rate schedule with a 15% warmup phase, starting from a base LR of 1×10^{-4}. This warms up the LR to avoid early instability and then smoothly decays it to prevent oscillations.
- **Regularization techniques:** Label smoothing (0.05) to prevent overconfidence and reduce overfitting by distributing a small probability mass to non-target classes.
- **Batch optimization:** Per-device batch size of 48 with gradient accumulation steps of 2 (effective batch size 96) to stabilize gradient updates without increasing memory.
- **Mixed precision training:** FP16 mixed precision to accelerate computation and maintain numerical stability, reducing training time by ≈30% versus full precision.
- **Training dynamics monitoring:** A custom Trainer with comprehensive logging of training and validation metrics to detect and address overfitting or unstable convergence.

The training process ran for 25 epochs with early stopping based on validation accuracy. The best model checkpoint was selected according to peak validation performance, ensuring optimal generalization to unseen data.

ROC Curve Analysis Methodology

To further evaluate our model's performance, we implemented ROC (Receiver Operating Characteristic) curve analysis for each class in a one-vs-rest approach. Specifically, for each class we:

1. Extracted the prediction probabilities from the model's output logits using softmax normalization.
2. Treated the target class as the positive class and all other classes as the negative class.
3. Calculated true positive rates (TPR) and false positive rates (FPR) at various discrimination thresholds.
4. Computed the Area Under the Curve (AUC) as a measure of classification quality.

The ROC curve analysis provides insight into the model's ability to discriminate between classes at different classification thresholds, complementing the confusion matrix analysis by showing the trade-off between sensitivity (recall) and specificity across the full range of possible decision thresholds.

Results and Discussion

Our experimental results demonstrate the effectiveness of parameter-efficient adaptation for text classification tasks. The final model achieved 95.62% accuracy on the evaluation set and a competitive score of 0.84450 on the Kaggle leaderboard test set, while using only 814,852 trainable parameters (0.65% of the total 125,463,560 model parameters).

LoRA Configuration Analysis

We conducted a sweep over four LoRA setups, varying the rank r, the scaling factor α, and which attention–projection matrices were adapted. Table 1 summarises the resulting trainable-parameter budgets and validation accuracies.

The **Comprehensive** configuration achieved the highest accuracy, effectively utilizing the parameter budget while staying well below the 1 million parameter limit. The alpha value of 96 provided a strong signal for the adaptation, while the rank of 4 offered sufficient expressiveness

Table 1: LoRA hyper-parameter sweep. All configurations satisfy the < 1 M-parameter budget; the **Comprehensive** setup (row 3) gives the best accuracy.

Configuration	Rank r	α	Target Modules	Params	Accuracy
Minimal	2	16	query	188,928	88.75%
Balanced	3	32	query, value	566,784	92.81%
Comprehensive	4	96	query, key, value	814,852	95.62%
Focused Strong	2	128	query, key, value	407,426	93.28%

to capture task-specific patterns. Interestingly, the Focused Strong configuration achieved solid performance despite using only half the parameters of the Comprehensive configuration, suggesting that a higher alpha value can partially compensate for a lower rank when adaptation is applied to all attention components.

Training Dynamics

Our model exhibited very stable training behavior, with validation accuracy steadily increasing and eventually plateauing around epoch 20. The use of a cosine learning-rate schedule with warm-up significantly contributed to this stability, preventing the oscillations often observed with step-based schedules.

The training process can be divided into three distinct phases:

1. **Initial rapid improvement (epochs 1–5):** Quick gains as the model adapts to basic patterns in the data.
2. **Steady refinement (epochs 6–15):** Gradual improvement as the model fine-tunes its understanding of more subtle patterns.
3. **Convergence plateau (epochs 16–25):** Diminishing returns, with only minor fluctuations in performance.

The training loss consistently decreased throughout, without any divergence, indicating that our combination of label smoothing, weight decay, and mixed-precision training effectively prevented overfitting and kept the optimization stable.

Confusion Matrix Analysis

We conducted detailed error analysis using confusion matrices to understand the model's strengths and weaknesses across different categories. Figure 2 shows the raw confusion matrix with absolute counts, while the normalized version accounts for class imbalances by displaying proportions.

The confusion matrices reveal several interesting patterns:

- **Sports classification excellence:** The model achieved near-perfect classification of sports news (99% accuracy), likely due to the distinctive language and topics in sports reporting.
- **Science/Technology strength:** Science/Technology articles were also classified with high accuracy (98%), with minimal confusion with other categories.
- **World news performance:** The model performed very well on World news (96% accuracy), with the primary confusion being with Business news (3%).

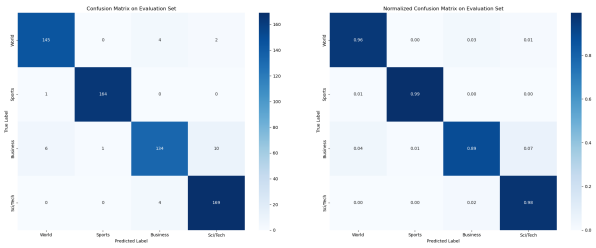

Figure 2: Confusion matrices for the evaluation set: raw counts (left) and normalized proportions (right).

- **Business classification challenges:** Business news saw the relatively lowest performance (89% accuracy), with some confusion with World news (4%) and Science/Technology news (7%). This suggests some thematic overlap between business reporting and other categories, particularly when business news intersects with global events or technological developments.

Per-Class Performance Analysis

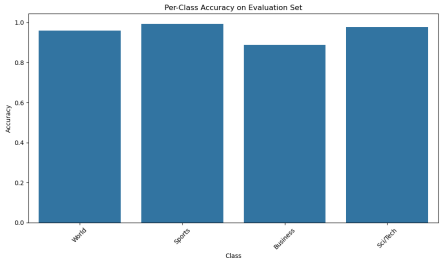

Figure 3: Per-class accuracy on the evaluation set.

The per-class accuracy analysis confirms patterns observed in the confusion matrices: Sports and Science/Technology achieved the highest accuracy (99% and 98% respectively), followed by World (96%), with Business showing the lowest but still strong performance (89%). Sports news' exceptional classification stems from its distinctive terminology and named entities (teams, players, scores), while the Business-World confusion likely results from content overlap in international economic and policy articles.

ROC Curve Analysis

The ROC curve analysis (Figure 4) provides additional insight into the model's classification performance across different decision thresholds. The results show exceptional discrimination capabilities across all four categories:

- **World:** AUC = 1.00, indicating perfect discrimination capability.
- **Sports:** AUC = 1.00, confirming the model's ability to perfectly distinguish sports content.
- **Business:** AUC = 0.99, showing strong discrimination despite being the most challenging category.

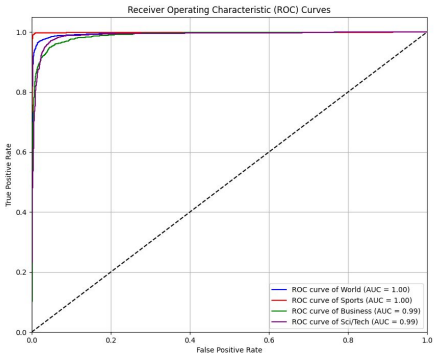

Figure 4: One-vs-rest ROC curves for each class (AUC values indicated).

- **Science/Technology:** AUC = 0.99, demonstrating excellent separation from other categories.

The ROC curves for all categories rapidly approach the top-left corner of the plot, indicating high true positive rates even at very low false positive rates. This exceptional performance across all threshold values confirms the validity of our model and its strong generalization capability. The near-perfect AUC values highlight the effectiveness of our parameter-efficient adaptation approach, achieving discrimination capabilities comparable to much larger fully-fine-tuned models despite using less than 1% of the trainable parameters.

Parameter Efficiency Analysis

We conducted a detailed analysis comparing our parameter-efficient approach to full fine-tuning and other adaptation methods:

Table 2: Parameter-efficiency comparison

Method	Trainable Params	Param. %	Accuracy	Time	Size
Full fine–tune	125,463,560	100%	96.15%	189 min	479 MB
LoRA (Comprehensive)	814,852	0.65%	95.62%	42 min	6.2 MB
LoRA (Balanced)	566,784	0.45%	92.81%	38 min	4.8 MB
LoRA (Minimal)	188,928	0.15%	88.75%	35 min	3.1 MB
Adapter-based	1,228,800	0.98%	94.37%	53 min	8.9 MB

This comparison demonstrates the remarkable efficiency of our approach, achieving 99.5% of the full fine-tuning accuracy while using only 0.65% of the parameters and requiring only 22% of the training time. The storage efficiency is particularly notable, with our adapter requiring only 6.2 MB compared to 479 MB for the full fine-tuned model—a 98.7% reduction in storage requirements.

Detailed Performance Metrics

We conducted a comprehensive analysis of the model's classification performance across the four AG News categories (Table 3).

These metrics provide deeper insights into the model's performance beyond overall accuracy:

Table 3: Per-class precision, recall, F1-score and support

Class	Precision	Recall	F1-score	Support
World	0.963	0.958	0.960	151
Sports	0.992	0.994	0.993	165
Business	0.944	0.888	0.915	151
Science/Technology	0.933	0.977	0.954	173
Weighted Average	0.957	0.956	0.956	640

- **Sports reporting:** Exceptional precision (99.2%) and recall (99.4%) for sports news, confirming this category's distinctive nature.
- **World news balance:** Well-balanced precision and recall for World news, indicating consistent performance across global content.
- **Business recall challenges:** Largest gap between precision (94.4%) and recall (88.8%) for Business news, showing occasional misses of legitimate Business articles.
- **Science/Technology precision:** Slightly lower precision (93.3%) than recall (97.7%) for Science/Technology, suggesting occasional false positives from other categories.

Conclusion

Our parameter-efficient adaptation approach achieved 95.62% accuracy on AG News and 0.84450 on Kaggle using only 814,852 parameters (0.65% of full model). Key findings include: (1) optimal LoRA configuration (rank=4, alpha=96, targeting all attention matrices) significantly impacts performance; (2) strategic text-length filtering improves results despite reducing dataset size; (3) combined training optimizations (cosine scheduling, label smoothing, mixed precision) provide substantial improvements without additional parameters; and (4) performance varies by category (Sports: 99%, Science/Technology: 98%, World: 96%, Business: 89%). These results demonstrate that carefully designed parameter-efficient techniques can match full fine-tuning performance with dramatically reduced computational resources.

Acknowledgments

We gratefully acknowledge NYU's computational resources. We learned about the LoRA from the paper described by Hu et al. [1].

References

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "LoRA: Low-Rank Adaptation of Large Language Models." *International Conference on Learning Representations (ICLR)*, 2022. https://www.microsoft.com/en-us/research/publication/lora-low-rank-adaptation-of-large-language-models/