News Classification with a Twist: DistilBERT and LoRA in a Low-Rank Tango

Yixuan Zhang
yiz119@ucsd.edu

1 Introduction

In this digital era filled with an overwhelming flood of information, countless news articles are being generated every day. Effectively and efficiently categorizing these news articles into specific and accurate categories, to help users quickly find the content they truly care about, is a significant challenge. Traditional text classification methods, such as Bag-of-Words and FastText, have addressed this problem to some extent, but as the complexity of information grows and classification demands diversify, these methods are increasingly falling short.

The goal of this project is to explore how to efficiently fine-tune DistilBERT using LoRA, in order to accurately classify news articles based on their descriptions, ultimately surpassing the performance of traditional text classification methods.

- **Dataset Preparation:** Collected and preprocessed the dataset. **(DONE)**
- **Baseline Model:** Built and trained the baseline model (using FastText) and evaluated its performance. **(DONE)**
- **Enhanced Model:** Implemented and trained the LoRA-enhanced DistilBERT model to improve classification performance. **(DONE)**
- **Model Comparison:** Compared and evaluated the performance of the LoRA-enhanced model with the baseline model. **(DONE)**

The experimental results show that fine-tuning DistilBERT with LoRA can significantly improve performance in the news classification task compared to the baseline FastText model.

2 Related Work

Text classification is a classic problem in natural language processing, and the field has seen significant advancements over the years. This section reviews the evolution of text classification methods in NLP, from traditional approaches to the latest Transformer-based methods.

2.1 Traditional Text Classification Methods

Early text classification methods relied on Bag-of-Words and simple neural network models. The Bagging method, proposed by Breiman (Breiman, 1996) at the University of California, Berkeley, generated multiple prediction models by bootstrapping the training set, significantly improving accuracy in text classification tasks. Even today, bagging remains relevant for text classification. However, these traditional methods lacked the ability to model semantic and contextual relationships within the text.

About a decade ago, pre-trained word embedding techniques such as Word2Vec and GloVe became the mainstream. By learning word context from large corpora, these methods enhanced text representation to some extent. However, as Word2Vec and GloVe relied on local window-based statistical information, they failed to capture word polysemy in different contexts, resulting in static word vectors that limited their performance.

2.2 The Rise of Transformer Models

The introduction of BERT in 2018 revolutionized text classification. As a Bidirectional Encoder Representation model based on Transformers, BERT (Devlin et al., 2018) enabled deep contextualized representations through its masked language model (MLM) pretraining objective. By eliminating the need for task-specific architectures, BERT allowed for simple fine-tuning while

achieving state-of-the-art results across multiple NLP tasks, including classification datasets.

Despite BERT's remarkable performance, its training process required substantial computational resources. RoBERTa (Liu et al., 2019) addressed this by optimizing BERT's pretraining process. It removed the "Next Sentence Prediction" (NSP) objective, increased batch size, and expanded training data, thus overcoming BERT's undertraining issues. RoBERTa further emphasized the importance of hyperparameter tuning and effective utilization of training data for better downstream task performance.

2.3 Efficient Transformer Models

As Transformer models evolved, their increasing computational costs posed challenges for real-world applications. DistilBERT (Sanh et al., 2019) tackled this issue by leveraging knowledge distillation during pretraining. It compressed the size of BERT by 40%, retaining 97% of its language understanding capabilities while increasing inference speed by 60%. DistilBERT's efficiency and performance made it an ideal choice for resource-constrained environments.

2.4 Fine-Tuning Strategies for Text Classification

As pre-trained models became central to NLP, fine-tuning strategies for text classification garnered significant research attention. Sun et al. (Sun et al., 2019) explored various fine-tuning methods to maximize BERT's potential in text classification tasks. They proposed techniques such as layer-wise learning rate adjustment and domain-specific pretraining. These methods achieved state-of-the-art results on several classification datasets, demonstrating that task-specific fine-tuning strategies can further enhance model performance.

2.5 Relevance to This Research

Building on the aforementioned studies, this project explores the combination of DistilBERT and Low-Rank Adaptation (LoRA) for efficient text classification. DistilBERT's computational efficiency, paired with LoRA's memory optimization, provides a promising solution for large-scale text classification tasks. This research aims to surpass traditional methods and existing pre-trained models by achieving higher accuracy and efficiency in categorizing news articles.

3 Dataset

3.1 Overview

The dataset used in this research is the AG News Classification Dataset, a widely used benchmark for text classification tasks. It consists of news articles categorized into four distinct classes based on their topics:

- **Class 1: World** (e.g., international news and global events)
- **Class 2: Sports** (e.g., sports events and athlete profiles)
- **Class 3: Business** (e.g., financial markets and economic trends)
- **Class 4: Sci/Tech** (e.g., scientific advancements and technology developments)

Each article contains a short title summarizing the content and a detailed description providing additional context. The classification task requires assigning the correct category to each description.

3.2 Data Source and Collection

The AG News Classification Dataset, used in this study, was constructed by Xiang Zhang (Zhang et al., 2015) for use as a benchmark for text classification tasks.

3.3 Dataset Statistics

- **Training Set:** 120,000 samples
- **Test Set:** 7,600 samples
- **Columns:**
  - **Class Index:** Integer labels (1–4) representing the class.
  - **Title:** A short summary of the article.
  - **Description:** A detailed description of the article content.
- **Text Characteristics:**
  - Average description length: ˜31 words (training), ˜30 words (test).
  - Vocabulary size: ˜160,000 unique words.

3.4 Sample Input-Output Pairs

- **Class 1: World** – "Reuters - South Korean police used water cannon in central Seoul Sunday to disperse at least 7,000 protesters urging the government to reverse a controversial decision to send more troops to Iraq."
- **Class 2: Sports** – "Reuters - The NFL Players Association Monday filed a grievance with the Dallas Cowboys over the recent release of quarterback Quincy Carter, claiming the organization was in violation of the league's collective bargaining agreement."
- **Class 3: Business** – "Reuters - Private investment firm Carlyle Group, which has a reputation for making well-timed and occasionally controversial plays in the defense industry, has quietly placed its bets on another part of the market."
- **Class 4: Sci/Tech** – "Reuters - A group of technology companies including Texas Instruments Inc. (TXN.N), STMicroelectronics (STM.PA) and Broadcom Corp. (BRCM.O), on Thursday said they will propose a new wireless networking standard up to 10 times the speed of the current generation."

3.5 Task Challenges

This task poses several challenges due to the nature of the dataset:

- **Semantic Overlap:** Certain descriptions, such as those related to business technology, may overlap between "Business" and "Sci/Tech" categories.
- **Text Length Variation:** Descriptions vary in length from concise summaries to verbose explanations, requiring flexible handling by the model.
- **Ambiguity in Context:** Subtle differences in descriptions (e.g., "market trends" vs. "scientific breakthroughs") necessitate deep contextual understanding.

3.6 Data Preprocessing

3.6.1 FastText Data Preprocessing

To align with the characteristics of the FastText model, the data preprocessing focuses on simplifying and standardizing the text. Specifically, all punctuation marks were removed, and the text was converted to lowercase to reduce the impact of case sensitivity on word distribution. Subsequently, based on the English stop word list provided by NLTK, high-frequency but semantically insignificant words (e.g., "the," "and") were removed to enhance the model's focus on key features. Finally, the cleaned text was formatted according to FastText's standard input format by prefixing each sample with a `__label__`tag, preparing the data for direct use in model training.

3.6.2 LoRA's DistilBERT Data Preprocessing

For the LoRA-enhanced DistilBERT model, the preprocessing strictly adheres to the input specifications of pretrained language models. First, the built-in tokenizer of the model was used to segment text into subword units, capturing fine-grained semantic information and fully leveraging the pretrained vocabulary. Subsequently, padding and truncation were applied to the tokenized results to ensure that all input samples had a uniform length (e.g., 128 tokens), thus enabling efficient batch processing and ensuring stability during training. This advanced preprocessing pipeline preserves semantic integrity while significantly improving the model's adaptability to variable-length text.

4 Baselines

The baseline model for this study is **FastText**, a lightweight and efficient text classification model designed to capture contextual information using word-level and n-gram representations. It serves as a robust benchmark for comparing the performance of advanced models.

4.1 Model Mechanism

FastText represents text as a bag of words and n-grams, allowing it to capture local word order and contextual patterns. By leveraging hierarchical softmax, the model is capable of efficiently handling multi-class classification tasks, even on large-scale datasets. Each input sample is annotated with a class label prefixed by `__label__`, ensuring compatibility with FastText's supervised training framework.

4.2 Hyperparameters and Tuning

The model's hyperparameters were optimized using FastText's built-in autotuning mechanism with a validation set. Key configurations include:

- **wordNgrams**: Set to 2 to capture bi-gram features, improving the model's contextual understanding.
- **autotuneValidationFile**: The validation set was provided to enable automatic hyperparameter tuning.
- **autotuneDuration**: Limited to 100 seconds to balance tuning quality and computational efficiency.

4.3 Train/Validation/Test Split

The training dataset was split into 90% for training and 10% for validation using stratified sampling to maintain class distribution consistency. The test set was held out and not used during hyperparameter tuning, ensuring an unbiased evaluation of the model.

4.4 Results

The FastText model achieved the following metrics on the test set:

- **F1 Score**: 0.9072

These results demonstrate the baseline model's effectiveness in handling text classification tasks and provide a solid benchmark for evaluating more complex models.

5 LoRA-Enhanced DistilBERT

5.1 Conceptual Approach

Our primary model is based on LoRA applied to a DistilBERT backbone for the classification task. LoRA introduces lightweight, trainable low-rank matrices into specific layers of the pretrained model, allowing efficient fine-tuning with minimal parameter updates. This approach reduces computational and memory overhead while retaining high performance.

LoRA adaptation is focused on the attention and feedforward layers, specifically targeting the modules `q_lin`, `v_lin`, `ffn.lin1`, and `ffn.lin2`. The key configuration parameters were set as follows:

- **Task Type**: Sequence Classification
- **Rank**: 8, balancing the expressiveness of low-rank matrices with computational efficiency.
- **Scaling Factor**: 32, to amplify the contributions of the low-rank matrices.
- **Dropout**: 0.2, to mitigate overfitting and enhance robustness.

5.2 Working Implementation

The LoRA-enhanced DistilBERT model is fully implemented, with the code organized into the following key components:

- **lora_finetune.py**: The main script orchestrating the pipeline, including data loading, model initialization, and training.
- **data_utils.py**: Handles data preprocessing tasks such as tokenization, label mapping, and input feature generation.
- **train_utils.py**: Implements training and evaluation logic, including optimization, loss computation, and metric reporting.

5.3 Compute

The experiments were conducted on a MacBook Pro equipped with an Apple M3 Pro chip (12-core CPU, 18-core GPU) and 18 GB of memory. The hardware proved sufficient for training the model efficiently, and no major computational issues were encountered.

5.4 Runtime

The training process consisted of three epochs, each taking approximately 26 minutes, resulting in a total training time of 1 hour and 18 minutes on the specified hardware.

5.5 Results

The LoRA-enhanced DistilBERT outperformed the baseline FastText model significantly. The results achieved on the test set are as follows:

- **Epoch 1**: Accuracy 91.96%, F1 Score 91.94%
- **Epoch 2**: Accuracy 92.55%, F1 Score 92.55%
- **Epoch 3**: Accuracy 92.82%, F1 Score 92.81%

In comparison, the baseline FastText model achieved an F1 score of 90.72%. These results highlight the effectiveness of LoRA in achieving high classification performance with minimal parameter updates.

5.6 Other Details

The LoRA configuration was carefully designed to balance computational efficiency and task-specific performance. The use of LoRA Dropout (set to 0.2) played a crucial role in preventing overfitting and improving the model's robustness to noise.

6 Error Analysis

6.1 Baseline Model: FastText

FastText relies on n-grams representation, which limits its ability to model long-range dependencies and contextual semantics. This leads to frequent errors in cases involving semantic ambiguity, complex sentence structures, and polysemous or rare words. For example, *"AI innovations in financial markets are driving growth"* is misclassified due to the overlap between "Sci/Tech" and "Business." Additionally, sentences with nested clauses, such as *"The player's strategy in esports reflects their understanding of AI models"*, confuse the model, resulting in classification errors as it struggles to distinguish between "esports" (Sports) and "AI models" (Sci/Tech).

6.2 Primary Model: LoRA-Enhanced DistilBERT

While LoRA-enhanced DistilBERT addresses many of FastText's limitations, it still struggles with semantically overlapping categories and rare or domain-specific terms. For example, *"Tech companies merge to strengthen their market position"* is misclassified as "Sci/Tech" instead of "Business" due to the emphasis on "tech companies." Similarly, *"Novel algorithms are being used to optimize market trading processes"* misclassified as "Business" instead of "Sci/Tech," as the term "market trading" aligns more closely with business terminology while ignoring the technical focus on algorithms.

6.3 Semantic and Syntactic Commonalities

These misclassified examples share significant semantic and syntactic commonalities. Most involve semantic ambiguity, such as terms spanning multiple categories or insufficient context to provide clear classification signals. Additionally, sentences with complex syntax, such as nested structures or multiple clauses, often lead to misclassification, especially in cases where categories exhibit high overlap.

7 Conclusion

This project demonstrated the efficiency of LoRA-enhanced DistilBERT for text classification under limited computational resources. With only 3 epochs of training, the LoRA fine-tuned DistilBERT model achieved an F1 score of 92.81% on the test set, representing a relative improvement of approximately 2.1% compared to the baseline FastText model's 90.72%. LoRA significantly reduced the computational cost of fine-tuning while achieving notable performance gains for the specific task. However, challenges remain in handling long texts and semantically overlapping categories. Future work will focus on leveraging larger pretrained models in environments with abundant computational resources to further enhance contextual understanding and address the limitations identified in this study.

References

Breiman, L. (1996). Bagging predictors. *Machine Learning*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? *arXiv preprint arXiv:1905.05583*.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*.