Model Comparison and Performance Optimization in News Text Classification

Zelin Zhang
Department of Electrical and Computer Engineering
Western University
London, Canada
zzha973@uwo.ca

Jian Li
Department of Electrical and Computer Engineering
Western University
London, Canada
jli4627@uwo.ca

Yinglun Suo
Department of Electrical and Computer Engineering
Western University
London, Canada
ysuo7@uwo.ca

Yanhua Zhang
Department of Electrical and Computer Engineering
Western University
London, Canada
yzha5778@uwo.ca

***Abstract*—AGnews is a common dataset in the field of NLP, and we hope to train an efficient news text classifier using AGnews. Due to its early publication, many authors have published training results on this dataset, but most of them are from several years ago. The development of deep learning models in the field of NLP has been very rapid in recent years, so I have decided to use more emerging models to train on this dataset and continuously adjust and optimize parameters, so as to train models with higher accuracy. In this task, we used transfer training to improve accuracy while reducing training time. Our model is divided into two groups. The first group uses the large size model to strive for the best performance. The second group is the small size group, exploring how to maximize the accuracy of the model with fewer parameters and computing resources. In this task, we successfully achieved higher performance or efficiency compared to older models.**

Keywords—Deep Learning, Natural Language Processing, Text Classification, Transfer Learning

I. INTRODUCTION

The rapid evolution of Natural Language Processing (NLP) technologies presents a unique opportunity to revisit the AGnews dataset [1]. It is a benchmark for news text classification, with modern deep learning models. Historically, studies utilizing AGnews have not tapped into the latest advanced models. These pre-trained models, developed through extensive training on diverse text data, have revolutionized understanding and processing of human language.

This thesis employs these state-of-the-art models via transfer training to significantly improve classification accuracy on the AGnews dataset. Our strategy involves using both large and small model variants to explore the balance between performance and computational efficiency. The aim is to not only achieve higher accuracy than previously reported but also to demonstrate the practical applicability of these advanced models in real-world scenarios, setting new standards for future NLP tasks.

The code of this project is now available. Please visit: https://github.com/zzha973/Group8_Project_2024_Section2_251376484_Code.git

II. BACKGROUND

In the field of Natural Language Processing (NLP), the exponential growth of deep learning methodologies has revolutionized text classification tasks, marking 2024 as a landmark year for innovation and advancements. This remarkable progress stems from a confluence of technological breakthroughs and theoretical insights, particularly in the development and refinement of transformer-based models, which have set new benchmarks for accuracy and efficiency. The surge in research output, characterized by a significant increase in published papers and deployed applications, underscores the field's rapid evolution. This year, advancements in pre-trained models such as BERT, RoBERTa, XLNet, ELECTRA, and DeBERTa have demonstrated unprecedented capabilities in understanding and processing human language, leveraging vast amounts of data and sophisticated training techniques. These models have not only surpassed previous records in text classification but also expanded the boundaries of what's achievable, enabling more nuanced and contextually aware interpretations of text. The progress in 2024 reflects a broader trend in deep learning, where NLP has emerged as a focal point of innovation, driving forward the capabilities of AI in understanding and interacting with human language.

Here are some state-of-the-art new technology which are used in this project:

A. Improvement in Hardware Acceleration

The development and training of state-of-the-art NLP models demand significant computational resources, particularly when processing large datasets like AGnews. In this study, we leverage NVIDIA's half-precision floating-point (FP16) acceleration technology available in the RTX 4090 graphics card. This technology significantly enhances the computational efficiency of our models, allowing for faster training times without compromising the accuracy of results. The FP16 acceleration is crucial for experimenting with large-scale models, facilitating rapid iteration and optimization processes [2].

B. Transfer Learning

Transfer learning has emerged as a transformative strategy in the field of machine learning, particularly in NLP. It involves the adaptation of models pre-trained on a broad dataset to

specific tasks, which in our case is news text classification [3]. This approach offers several benefits:

1) Efficiency: Transfer learning dramatically reduces the need for large amounts of labeled data, which is often a bottleneck in NLP tasks.

2) Improved Performance: Leveraging models pre-trained on extensive datasets provides a solid foundation, often leading to enhanced model performance on specific tasks.

3) Speed: It significantly accelerates the training process, making it feasible to deploy more sophisticated models within practical timeframes.

C. Several State-of-the-art models

Our research utilizes five cutting-edge pre-trained models, each contributing unique strengths to the task of text classification:

1) BERT (Bidirectional Encoder Representations from Transformers): Developed by Google, BERT revolutionized the understanding of context in text by processing words in relation to all other words in a sentence, rather than in isolation [4].

2) RoBERTa (A Robustly Optimized BERT Pretraining Approach): An iteration on BERT, RoBERTa optimizes the pre-training process, leading to improved model performance and efficiency [5].

3) XLNet: Utilizing a permutation-based training strategy, XLNet outperforms BERT in several benchmarks by capturing the bidirectional context of data more effectively [6].

4) ELECTRA: Instead of the traditional masked language model training, ELECTRA employs a generator-discriminator setup that trains more efficiently, providing a competitive edge in terms of performance and speed [7].

5) DeBERTa (Decoding-enhanced BERT with Disentangled Attention): Introducing a disentangled attention mechanism, DeBERTa improves the representation of word sequences, achieving remarkable results on various NLP tasks [8].

Each of these models brings forth advancements in language processing capabilities, offering nuanced understanding and contextual interpretation of text. By applying these models in conjunction with NVIDIA's FP16 acceleration on the RTX 4090, this thesis aims to set new benchmarks in the efficiency and accuracy of news text classification.

D. AdamW Optimizer

In the landscape of deep learning optimization algorithms, the AdamW optimizer emerges as a refined variant of the widely acclaimed Adam optimizer, integrating the concept of weight decay directly into the optimization process. Unlike traditional weight decay, which is applied as a separate step after the update, AdamW incorporates weight decay into the optimizer's update rule itself [9]. This distinction addresses the issue of L2 regularization not behaving as true weight decay in Adam, particularly when adaptive learning rates are involved. By decoupling the weight decay from the optimization steps, AdamW promotes a more effective regularization method, leading to improvements in training stability and model generalization. This is especially crucial in complex NLP tasks, where the balance between learning complex representations and avoiding overfitting on the training data is paramount. Utilizing AdamW in the training of advanced NLP models, such as those employed in this thesis, leverages these benefits, potentially enhancing model performance on tasks like news text classification.

III. Methodology

A. Dataset description and processing

1) Analysis of AGnews Dataset: AGnews dataset consists of news articles categorized into four main classes. We also analyze the top 50%, 90%, 99% length of the items in the

2) dataset to ensure the length of tokenizer. Finanly the length of token was fixed to 90.

3) Data Preprocessing: We employed standard NLP preprocessing techniques, including tokenization using BertTokenizer and RobertaTokenizer from the Hugging Face library. This step converts text data into a format suitable for model input, ensuring that it captures the textual information effectively for classification tasks.

B. Model Selection and Rationale

1) Review of Existing Models: Previous studies have applied various models to the AGnews dataset, ranging from early machine learning algorithms to more recent deep learning approaches.

2) Rationale for Model Selection: Our choice to utilize BERT and RoBERTa models stems from their state-of-the-art performance in various NLP tasks. Their pre-trained representations, capable of capturing deep contextual relationships in text, offer a promising foundation for achieving high accuracy in news classification.

C. Training Process

1) Hardware and Software Environment: Training was conducted on GPU-accelerated hardware, utilizing PyTorch as the primary deep learning framework. Mixed precision training was implemented using PyTorch's autocast and GradScaler for efficient memory usage and faster computation.

2) Training Procedure: Models were fine-tuned using the AdamW optimizer, with a learning rate scheduler to adjust the rate throughout training. Training involved processing data in batches, with size and epochs adjusted to optimize performance.

3) Performance Metrics: Accuracy was the primary metric for evaluating model performance, calculated by comparing the model's predictions against the true labels of a validation set.

D. Testing and Evaluation Process

1) Accuracy Measurement: The primary metric used to evaluate the model is accuracy, which is calculated by the compute_accuracy function. This function assesses how often the model's predictions match the true labels of the news articles in the validation dataset. Accuracy is a straightforward and intuitive metric, making it suitable for a first-level evaluation of the model's performance.

2) Continuous Evaluation: At the end of each training epoch, the model is switched to evaluation mode (model.eval()) to test its performance on the validation dataset. This switch disables certain layers and behaviors, such as dropout, that are only relevant during training, ensuring that the model's performance is assessed under conditions that mimic its eventual use.

3) Best Accuracy Reporting: The highest recorded validation accuracy is reported at the end of training, summarizing the effectiveness of the training process and the model's capability in classifying unseen news articles.

E. Computational Resources

The code is designed to run on a GPU NVIDIA RTX4090, which has 24GB VRAM, leveraging CUDA for accelerated training. Then, we implement have-precision float(FP16) to accelerate the training process. Then, we This highlights the computational considerations and optimizations made for efficient model training.

IV. EXPERIMENTAL IMPLEMENT

A. Explanation of hyperparameters

1) Find the best token length: In order to maximize our model performance and reduce training time, we need to choose a token length to ensure that this length can basically cover the items in the dataset. Reduce the potential loss of information without causing excessive computational resource waste.

The result is shown at table1.

TABLE 1. THE LENTH DISTRIBUTION IN DATASET

	Average	***Median***	***90%***	***99.5%***	***99.9%***
percentage	38	37	48	80	112

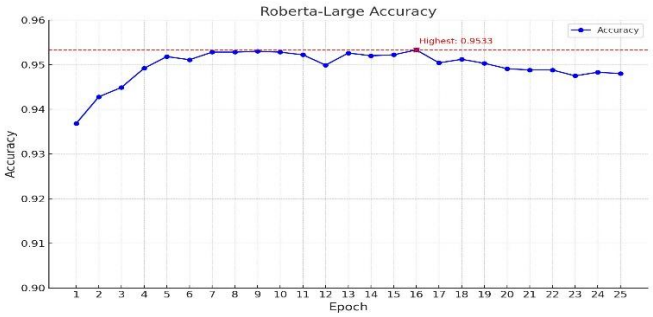
Based on this result, we have decided to choose an interval between 99.5 and 99.9 as our token length, which can retain most of the information, reduce information loss in data preprocessing, and prevent tokens from being too large and reducing performance. We ultimately chose 90 as the standard for our token length.

2) Adjustment of hyperparameters for Dataloder: Due to the use of RTX4090, a powerful GPU with 24GB VRAM, and the use of FP16 to train our model, we were able to use a larger batch size on the dataloader. The virtual machine instance we rented has 8 CPU cores, so we set the number of threads called by the dataloader to 8. After verification, our GPU core usage has been consistently maintained at over 95%, with a graphics memory usage rate of approximately 21GB/24GB, fully leveraging the performance of our hardware.

3) Epochs and learning rate: We used early stop technique to prevent the model from deteriorating performance due to oversaturation. After each round of training, we will verify the performance on-site in the validation set. Only when the accuracy improves, the model will be saved as a. pt file to ensure that the saved model is the strongest throughout the entire training cycle.

Here is an example of the learning period:

FIGURE 1. ACCURACY TREND OF ROBERTA(LARGE SCALE)



We can know that after epoch16, the overfitting appears and the accuracy of the model in validation set continues to decrease. We will stop saving the model(.pt file) after epoch16 to ensure that we save the best model.

B. Software and Hardware environment

1) Hardware environment: Due to the maximum model parameter being above 300M, we have decided to lease a remote server. We use a virtual machine with an RTX4090 GPU as the experimental equipment, and in addition, the VM is equipped with 8 CPU cores. In theory, this GPU has a single precision 82.58 TFLOPS or a semi precision 165.2 Tensor TFLOPS.

2) Software version

TABLE 2. SOFTWARE VERSION

PyTorch	2.1.0
Python	3.10
Cuda	12.1
OS	ubuntu22.04

V. EXPERIMENTAL RESULT

A. Comparison between single precision floating-point(FP16) and double precision floating-point(FP32).

We use BERT-base model as the reference when we compare both of these two float precision.

TABLE 3. MODEL ACCURACY AND TRAINING TIME OF FP32 AND FP16

loat precision	FP16	FP32
Max accuracy	0.9434	0.9433
Time of each epoch	1min 41sec	2min 57sec

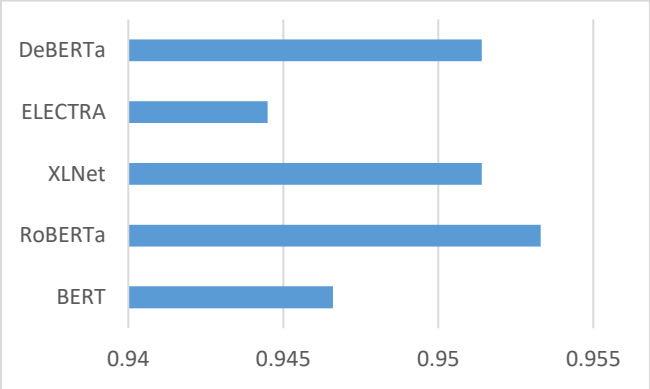
We can easily observe that when using semi precision floating-point (FP16) as the training accuracy, the accuracy of the model is almost no different from using single precision floating-point (FP32), but the training time is significantly reduced. We used FP16 as the training accuracy for the model in all subsequent experiments.

B. Large scale group accuracy comparison

All of them were proposed in the past three years and have been validated in other datasets, proving to have achieved good results. To control variables, we all use pre trained models(Large-scale version) integrated in the Transformer package. Comparing under the same hyperparameters and system environment

1) The accuracy of five large scale models

FIGURE 2. ACCURACY OF EACH MODEL(LARGE SCALE GROUP)



We can easily find that the RoBERTa model got the best performance in this round

2) Training time of five large scale models

TABLE 4. COMSUPTION OF TRAINING TIME FOR EACH EPOCH (RTX4090)

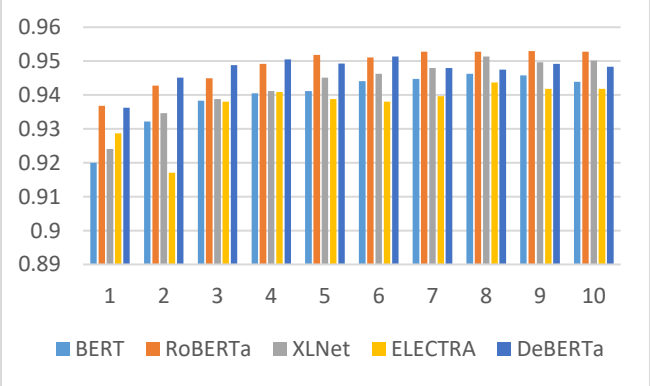
BERT	RoBERTa	XLNet	ELECTRA	DeBERTa
4min 26sec	4min 30sec	6min 16sec	4min 33sec	6min 49sec

We can know that in this round, RoBERTa get the second ranking while BERT get the best.

3) Convergence speed of five large scale models

We cut the accuracy after first ten epochs' training

FIGURE 3. ACCURACY AFTER FIRST TEN EPOCHS(LARGE GROUP)



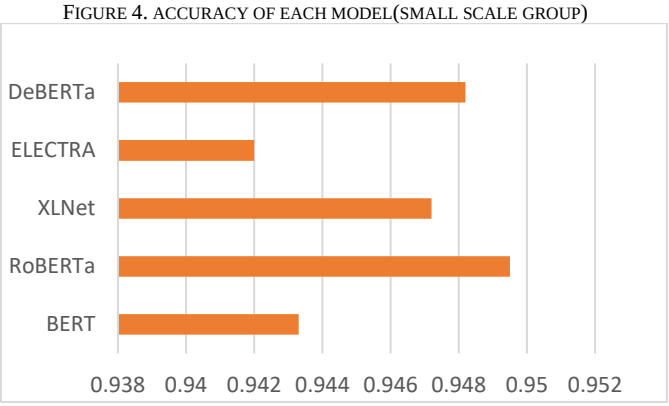
We can find that RoBERTa and DeBERTa got the best result in this round because they

C. small scale group accuracy comparison

All environment variables are exactly the same as when training on the large scale earlier. Change the model from the large version pre trained model integrated in the dataset package to its corresponding small scale pre trained model.

1) The accuracy of five small scale models

FIGURE 4. ACCURACY OF EACH MODEL(SMALL SCALE GROUP)



Just like the situation in the large scale version. We can easily find that the RoBERTa model got the best performance in this round.

2) Training time of five large scale models

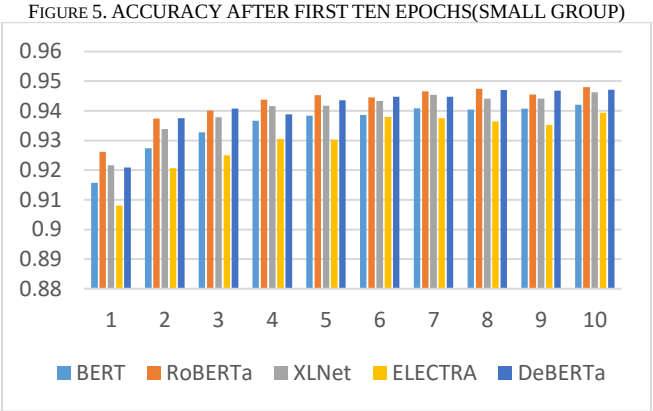
TABLE 5. COMSUPTION OF TRAINING TIME FOR EACH EPOCH (RTX4090)

BERT	RoBERTa	XLNet	ELECTRA	DeBERTa
1min 41sec	1min 34sec	2min 25sec	1min 36sec	2min 33sec

We can know that in this round, RoBERTa is the best model in time efficiency.

3) Convergence speed of five large scale models

FIGURE 5. ACCURACY AFTER FIRST TEN EPOCHS(SMALL GROUP)



VI. CONCLUSION

We can easily draw conclusions by comparing all the experimental results. RoBERTa is the most suitable dataset among several modern advanced models. Whether in a 300M parameter level large model for performance comparison or a 100M parameter level base model. RoBERTa Large has become the model with the highest performance, fastest convergence, and the second shortest training time per epoch. In the basic model group, we found that RoBERTa Base achieved the best performance in all three important indicators: accuracy, convergence speed, and training time per epoch. This fully demonstrates the strong ability of RoBERTa in the task of news text classification, making it the most competitive model structure in this sub field.

ACKNOWLEDGMENT

All members of author group of this project would like to express our gratitude to Dr. Ahmed Ibrahim, the instructor of the ECE9039 machine learning course for his sincerest gratitude for the knowledge he has taught and guidance on this project.

REFERENCES

[1] Xiang Zhang, Junbo Zhao and Yann LeCunJ, "Character-level Convolutional Networks for Text Classification" Advances in Neural Information Processing Systems 28 (NIPS 2015)

[2] NVIDIA. "NVIDIA Ada GPU Architecture." [Online]. Available: https://images.nvidia.com/aem-dam/Solutions/geforce/ada/nvidia-ada-gpu-architecture.pdf&#8203K.

[3] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186, Jun. 2019.

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining

[6] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding.

[7] Clark, K., Luong, M.-T., Khandelwal, U., Manning, C. D., & Le, Q. V. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.

[8] He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention.

[9] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2019. [Online]. Available: https://ar5iv.labs.arxiv.org/html/1711.05101