Parallelizing BERT-based Detection of Fake News in the Filipino Language

Mary Nathalie Dela Cruz

Artificial Intelligence Department, College of Engineering, University of the Philippines Diliman
marynathaliedelacruz@gmail.com

Abstract

Fake news, defined as deliberately false information crafted as credible news, poses significant threats to society by misleading and deceiving the public. Although traditional manual fact-checking methods are effective, they are limited in practicality and scalability. This study evaluated the effectiveness of a BERT-based model for fake news detection in the Filipino context, leveraging GPU parallelism to handle computational demands. Using a localized dataset of 1603 Filipino news articles that are either fake or true, the study investigated the impact of various training parameters on model performance. We highlighted that an optimal configuration with a batch size of 32, two data loaders, and a learning rate of 0.00006 with two training epochs, can achieve high accuracy and efficient training time. This study can contribute to enhanced fake news detection capabilities in the Philippines.

Keywords: fake news, NLP, BERT, machine learning, GPU

1 Introduction

Fake news refers to verifiably false information intentionally crafted as credible news aiming to mislead and deceive. These deceptive practices negatively impact the public and the news ecosystem as they can influence public opinion, incite political instability, and undermine societal trust. As such, with the continuous rise of fake news, particularly on social media platforms, robust and effective strategies for fake news detection and mitigation must be developed. Online fact-checking systems, such as FactCheck.org and PolitiFact.com, traditionally relied on manual verification by experts. However, the practicality and scalability of those systems are limited because of time latency, and high veracity, variety, and volume of fake news [4]. To address these limitations, automatic detection techniques with improved generality and performance have been developed, ranging from simple keyword matching to machine learning algorithms [1-5].

A machine learning-based fake news detection typically involves training models on datasets to identify patterns indicative of fake news. These patterns can be based on features from users, creators, news content, and news propagation [4]. Recently, transformer models like BERT (Bidirectional Encoder Representations from Transformers) have made a breakthrough in this domain. BERT is designed to enable deep bidirectional (left and right) text representation, allowing it to recognize and capture the context of a text [6-9]. However, training and inference of BERT models are computationally intensive. Graphics Processing Units (GPUs) offer a solution by providing computational resources capable of handling massively parallel processing tasks [8].

In this study, we aim to evaluate the improvement of a BERT-based fake news detection system through GPU utilization. Because research on fake news detection in the Philippines is still underdeveloped, a localized labeled dataset relevant to the Filipino context was used for training. Existing studies have focused on curating datasets of Filipino fake news articles and developing initial models for detection. One study even explored the effectiveness of transformer-based models, highlighting their potential in this field [9-10]. By implementing GPU-accelerated BERT models in the Filipino context and evaluating different model parameters, we aim to contribute to the fight against the spread of fake news in the Philippines.

2 Preliminaries

In this study, the dataset curated by Cruz et al was used for training the BERT-based model for fake news detection [9]. This dataset is a collection of 1603 news articles in the Filipino language. Half were labeled as fake news and

the other half was true news. Figure 1 shows the word cloud for each label without the stop words, where higher-frequency words have larger font sizes.

Figure 1: Word cloud of text labeled as fake news and true news

Figure 2 shows the text length distribution for each label. The text collection labeled as fake news has a maximum length of 1074 and a mean length of 121. The text collection labeled as true news has a maximum length of 965 and a mean length of 244.

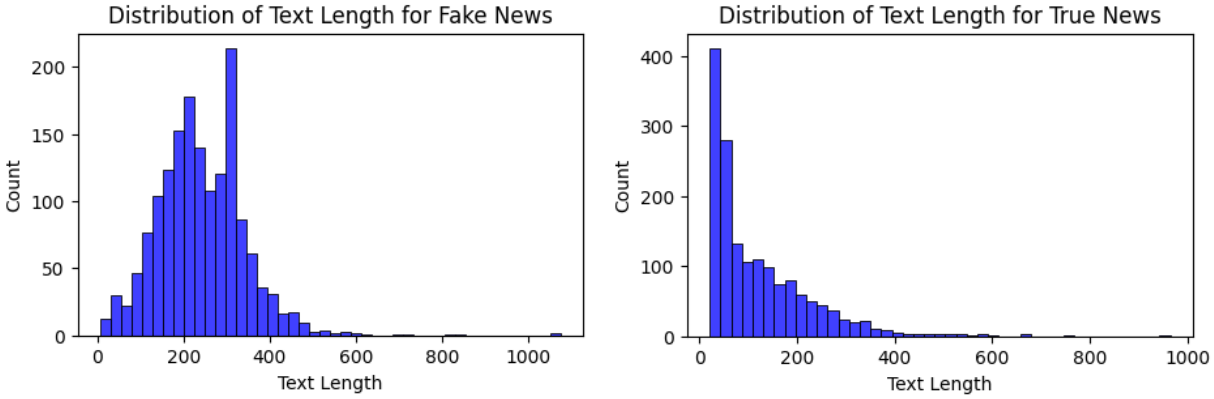


Figure 2: Length distribution of text labeled as fake news and true news

NVIDIA-L4 GPU from Google Colab was employed for training and evaluating the BERT-based models. Unlike traditional CPUs, the NVIDIA-L4 GPU can handle the computational demands of BERT models, even for large datasets and computational tasks. Table 1 shows the Cuda device properties obtained from PyTorch library [11].

Table 1: Cuda Device Properties

Name	NVIDIA-L4
Major	8
Minor	9
Total Memory (mb)	22699
Multi Processor Count	58

The model used to develop the fake news detection system was BERT (Bidirectional Encoder Representations from Transformers). BERT is a deep learning model that uses transformers to understand the word context by pre-training bidirectional representation. The main pre-training tasks were Masked Language Model (MLM), which predicts masked tokens in the input sequence to learn context from left and right directions, and Next Sentence Prediction (NSP), which predicts the next sentence to understand the relationship between sentence pairs. Its architecture is based on the Transformer model of a specific number of layers, hidden size, and attention heads. After pre-training, BERT can be fine-tuned with minimal modifications [7].

3 Methodology and Related Works

The training and evaluation of a BERT-based fake news detection system are as follows. First, the availability of the GPU was checked and the device to be used in the workflow was set accordingly. For this system, the device was set to Cuda. The dataset curated by Cruz et al was downloaded from Hugging Face and was split into features (X), which contains the articles, and labels (y), where "0" represents true news and "1" represents fake news [12]. BERT tokenizer was loaded and used to encode the dataset, where each text was converted into smaller parts called tokens [13]. The encoded data and the corresponding label were modified into a tensor dataset by a custom dataset class which was then fed into the model. Model performance was evaluated by iterating over different parameter configurations while performing k-fold cross-validation for 3 folds. The parameters investigated in this study were batch size, number of DataLoader workers, number of epochs, and learning rate [14-15]. The model was trained and evaluated for each parameter configuration and each fold. Performance metrics such as training time, accuracy, f1 score, precision, and recall were recorded. The script can be viewed in Colab [19].

This study is similar to the research made by Cruz et al [9]. Both used BERT-based models and focused on localized datasets in the Filipino language for fake news detection. However, the approaches and objectives of these studies diverge. This study utilized GPU parallelism to enhance training efficiency, exploring various training parameters to optimize model performance. We focused on reducing the computational demands of BERT to make real-time fake news detection more feasible. In contrast, the research made by Cruz et al leveraged transfer learning and multitask learning to improve performance with limited data. Their project managed to mitigate data scarcity and enhance model adaptability. Together, these studies contribute insights into improving fake news detection in the Filipino context by balancing computational efficiency and model robustness.

4 Results and Discussion

Figure 3 shows the impact of batch size on training time and model accuracy. Batch size refers to the number of samples processed by GPU. Note that tensor core requirements define the optimal batch size. Given that the model was trained on a 16-bit floating point precision in NVIDIA-L4 with CUDA version 12.2, the batch size set was decided to be multiples of 8 [13]. As shown below, a larger batch size shortens the training time because it improves gradient estimation, leading to faster convergence. However, a larger batch size may lead to poorer generalization due to reduced noise in gradient estimation, leading to a lower model accuracy [14-17].

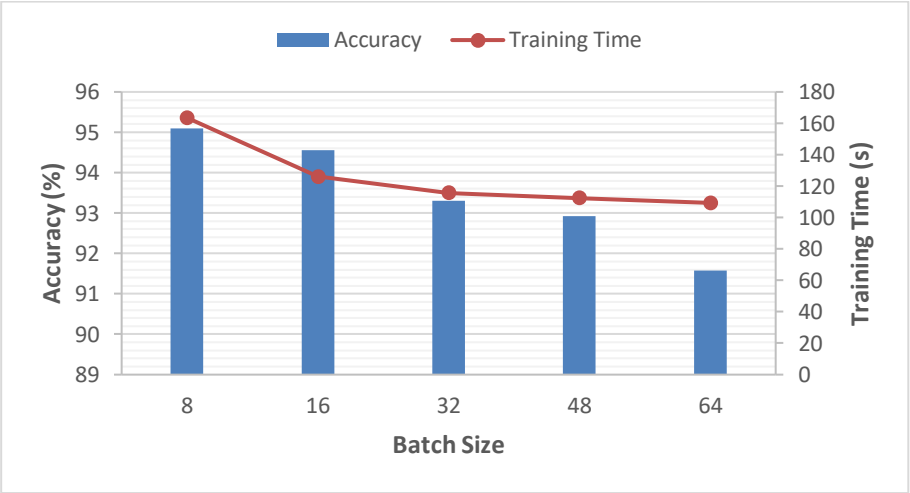
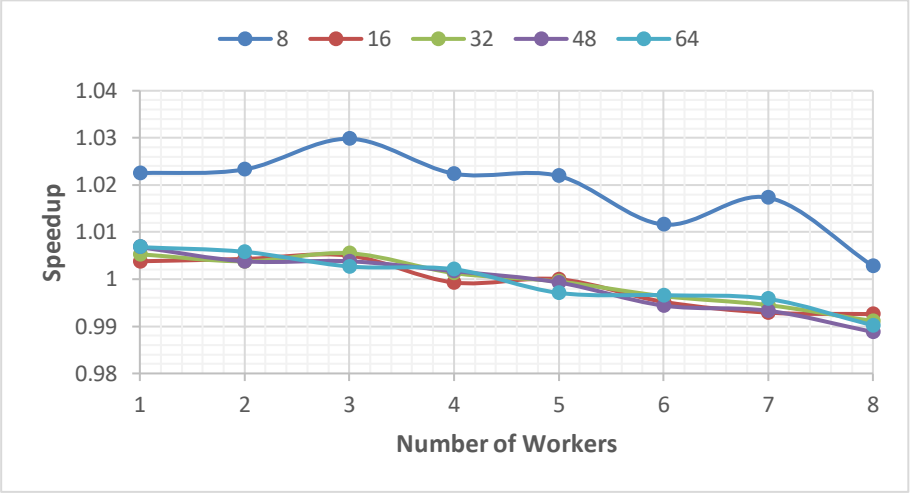


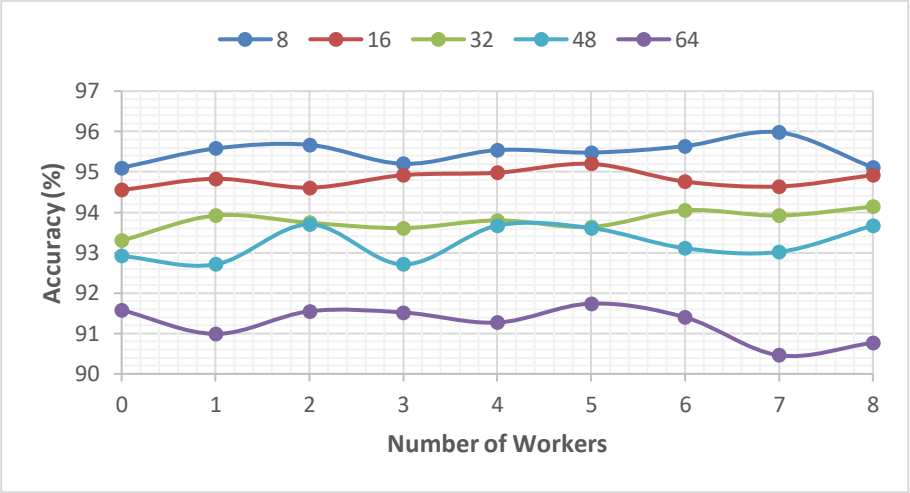
Figure 3: Impact of batch size on training time and accuracy of the model

In data loading, multiple workers can load a batch of data in parallel. Hence while the model is processing a batch, other workers can load the next batch. As shown in Figure 4a, more workers speed up the data loading and reduce the time the model waits for the data. This is observed until the number of workers was set to 4. Beyond that, the GPU is already utilized to its maximum capacity and no further speedup can be observed, leading only to an

increase in memory usage. Figure 4b shows that the number of data loaders does not inherently affect the model accuracy [14-16].



(a) Speedup vs number of data loaders



(b) Accuracy vs number of data loaders

Figure 4: Impact of number of workers in data loading to (a) speedup and (b) model accuracy

Speedup is observed in the batch size set when there are 2 or 3 data loaders. Figure 5 further shows the effect of 2 and 3 data loaders on accuracy and training time for the batch size set. Considering the tradeoffs between accuracy, training time, and memory usage, the optimal batch size was 32 with 2 data loaders.

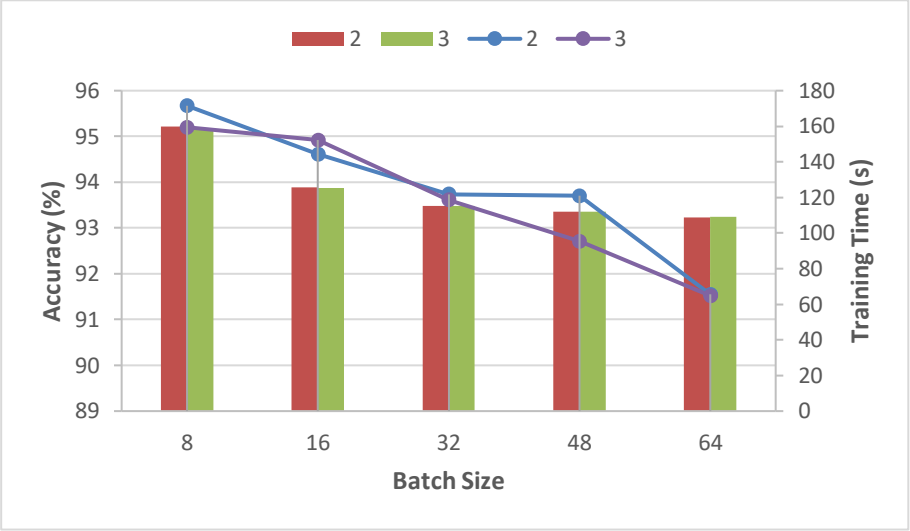


Figure 5: Impact of 2 and 3 data loaders on accuracy and training time

An epoch entails a complete pass through the training dataset. With every epoch, the model parameters are updated based on the data where fewer epochs increase the risk of underfitting while more epochs increase the risk of overfitting. Hence, finding the number of epochs that balances model accuracy and training time is necessary. Figure 6 illustrates how the number of epochs affects the model accuracy and training time. It shows that model accuracy increases until 3 and 4 epochs, where the model has reached a bottleneck. On the other hand, training time increases linearly with the number of epochs. From this figure, 3 training epochs are optimal [16-18].

Figure 6: Impact of number of epochs on accuracy and training time

The learning rate controls the step size of the model optimization. It affects the precision of convergence where lower learning rates lead to slower convergence, longer training time, and a higher risk of getting stuck in the local minima, meanwhile, higher learning rates lead to faster convergence, shorter training time, and a higher risk of overshooting the local minima. Typically, higher learning rates require fewer training epochs. Figure 5 helps us choose an optimal learning rate with the right number of epochs considering model accuracy and training time. Note that for this study, the training time of varying learning rates is close and averages to the training time that linearly increases with the number of epochs. Based on this figure, the optimal learning rate is 0.00006 with 2 training epochs [16-18].

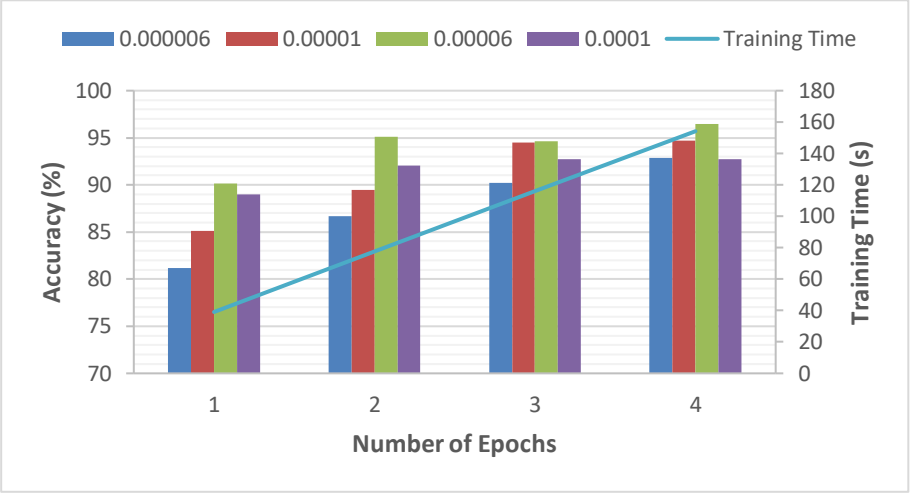


Figure 7: Impact of learning rate and number of epochs on accuracy and training time

From the experiment, we can say that an optimal BERT-based fake news detection model ran on NVIDIA-L4 GPU with the Filipino dataset has a batch size of 32 with 2 data loaders, and a learning rate of 0.00006 with 2 training epochs. Table 2 shows the average performance metrics of this model in 3 k-fold cross-validation. The model obtained an accuracy of 94.39% with an average training time of 76.26 seconds per epoch.

Table 2: Performance Metrics

Training Time (s)	76.26
Accuracy	0.9439
F1 Score	0.9440
Precision	0.9417
Recall	0.9463
Validation Loss	0.1645

5 Conclusion

The study successfully demonstrated the effectiveness of GPU-accelerated BERT-based models in fake news detection within the Filipino context. By optimizing training parameters such as batch size, number of data loaders, number of epochs, and learning rate, the model achieved an accuracy of 94.39% with an average training time of 76.26 seconds per epoch. This method offers a scalable and efficient solution to combat the spread of fake news in the Philippines. Further research can explore parameters/techniques such as gradient accumulation, gradient checkpointing, and optimizer choice to enhance model performance and GPU resource utilization [14].

6 References

[1] Lazer, David MJ, et al. "The science of fake news." Science 359.6380 (2018): 1094-1096.

[2] Shu, Kai, et al. "Fake news detection on social media: A data mining perspective." ACM SIGKDD explorations newsletter 19.1 (2017): 22-36.

[3] Sharma, Uma, Sidarth Saran, and Shankar M. Patil. "Fake news detection using machine learning algorithms." International Journal of creative research thoughts (IJCRT) 8.6 (2020): 509-518.

[4] Zhang, Xichen, and Ali A. Ghorbani. "An overview of online fake news: Characterization, detection, and discussion." Information Processing & Management 57.2 (2020): 102025.

[5] Khanam, Zeba, et al. "Fake news detection using machine learning approaches." IOP conference series: materials science and engineering. Vol. 1099. No. 1. IOP Publishing, 2021.

[6] Kula, Sebastian, Michał Choraś, and Rafał Kozik. "Application of the BERT-based architecture in fake news detection." 13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020) 12. Springer International Publishing, 2021.

[7] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[8] Lin, Jiahuang, Xin Li, and Gennady Pekhimenko. "Multi-node Bert-pretraining: Cost-efficient approach." arXiv preprint arXiv:2008.00177 (2020).

[9] Cruz, Jan Christian Blaise, Julianne Agatha Tan, and Charibeth Cheng. "Localization of fake news detection via multitask transfer learning." arXiv preprint arXiv:1910.09295 (2019).

[10] Fernandez, Aaron Carl T. "Computing the Linguistic-Based Cues of Credible and Not Credible News in the Philippines Towards Fake News Detection." (2019).

[11] NVIDIA. "NVIDIA L4 Autonomous Vehicle Development Platform." NVIDIA, n.d., https://www.nvidia.com/en-us/data-center/l4/.

[12] Hugging Face. "Fake News Filipino." Hugging Face Datasets, n.d., https://huggingface.co/datasets/jcblaise/fake_news_filipino.

[13] NVIDIA. "Tensor Cores: Matrix Multiplication Performance Guide." NVIDIA Deep Learning Performance Documentation, n.d., https://docs.nvidia.com/deeplearning/performance/dl-performance-matrix-multiplication/index.html#requirements-tc.

[14] Hugging Face. "Methods and Tools for Efficient Training on a Single GPU." Hugging Face Transformers Documentation, n.d., https://huggingface.co/docs/transformers/perf_train_gpu_one#methods-and-tools-for-efficient-training-on-a-single-gpu.

[15] Hugging Face. "Trainer." Hugging Face Transformers Documentation, n.d., https://huggingface.co/docs/transformers/main_classes/trainer.

jcblaise/fake_news_filipino · Datasets at Hugging Face

[16] NVIDIA. "Tensor Cores: Matrix Multiplication Performance Guide." NVIDIA Deep Learning Performance Documentation, n.d. Web. https://docs.nvidia.com/deeplearning/performance/dl-performance-matrix-multiplication/index.html#requirements-tc.

[17] Smith, Leslie N. "A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay." arXiv preprint arXiv:1803.09820 (2018).

[18] Shafi, Sadaf, and Assif Assad. "Exploring the Relationship Between Learning Rate, Batch Size, and Epochs in Deep Learning: An Experimental Study." Soft Computing for Problem Solving: Proceedings of the SocProS 2022. Singapore: Springer Nature Singapore, 2023. 201-209.

[19] Google. "CS 239 Parallel Programming.ipynb" (Notebook). Google Colab, n.d. Web. https://colab.research.google.com/drive/1bLiMBwVqGDzrQkm-P5lzsO5_T6D_1XNQ?usp=sharing.