Fine-Tuning BERT for Fake News Detection

1st Yejin Park
Department of Statistics
University of Michigan
Ann Arbor, MI, USA
yejin@umich.edu

***Abstract*—This paper presents a BERT-based approach for fake news detection using the WELFake dataset. By leveraging preprocessing techniques and fine-tuning, the model achieved a validation accuracy of 92.09% and a test accuracy of 92.73%. These results demonstrate the effectiveness of BERT in addressing the challenges of misinformation in digital spaces.**

***Index Terms*—BERT, fake news detection, fine-tuning**

I. Introduction

Artificial intelligence is automating the creation of fake news, spurring an explosion of web content mimicking factual articles that instead disseminates false information about elections, wars and natural disasters. Since May 2023, websites hosting AI-created false articles have increased by more than 1,000 percent, ballooning from 49 sites to more than 600, according to NewsGuard, an organization that tracks misinformation [1]. To address this issue, I aim to develop a fake news detection model using Bidirectional Encoder Representations from Transformers (BERT) and analyze its process.

II. Related Work

Kaliyar et al. [2] proposed a BERT-based deep learning approach called FakeBERT, which combines BERT with parallel blocks of single-layer CNNs with varying kernel sizes and filters. This model achieved a superior accuracy of 98.90%, outperforming existing models in fake news detection. Kula et al. [3] emphasized the need for flexible and intelligent tools in modern computer security systems to combat fake news, highlighting the role of advanced neural network architectures like BERT. The study proposed a hybrid architecture combining BERT with Recurrent neural networks (RNN), designed for effective fake news detection.

III. Methodology

The process of fake news detection can be devided into four stages – Data Overview, Data Preprocessing, Sentence Embedding, and BERT Model Fine-Tuning.

A. Dataset Overview

The dataset I use is the WELFake dataset [4] contains a total of 72,134 news articles, balanced between 35,028 real and 37,106 fake news samples.

To get insight into the dataset, I visulaized histograms for the token distributions of both titles and full-text data. Token calculations were conducted by estimating the number of tokens as approximately 1.3 times the word count. [7]

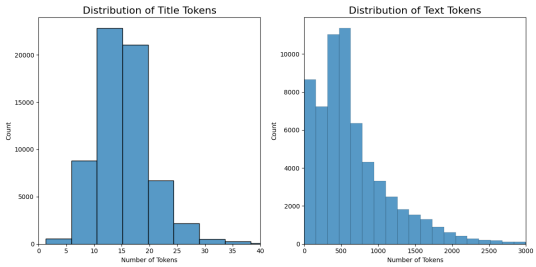

Fig. 1. Histogram of Estimated Token Counts for Titles and Text Data

I selected the title data for model training because its token distribution is more concentrated and narrower compared to full-text data, making it more computationally efficient.

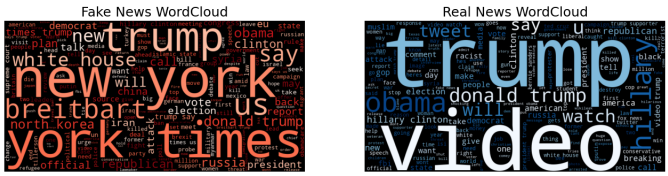

Fig. 2. Word Cloud for News Titles

We can see that, in the fake news word cloud, terms like “New York,” “Trump,” and “Times” appeared frequently, indicating a focus on specific locations and individuals often used to enhance credibility. On the other hand, in the real news word cloud, words such as “Trump,” “Video,” “Obama,” and “Watch” appeared most frequently, suggesting an emphasis on multimedia content and prominent political figures. From these observations, the word clouds provide some important insights to differentiate between fake and real news articles.

B. Data Preprocessing

To prepare the title data, a custom text cleaning function was implemented. The function performs the following steps:

- Removes special characters, retaining only alphanumeric characters and spaces.

- Converts all text to lowercase to ensure consistency across the dataset.

This step ensures that unnecessary noise is eliminated, making it more suitable for training a transformer-based model.

C. Sentence Embedding

To process the titles in the dataset for input into the BERT model, I utilized the pre-trained `bert-base-uncased` tokenizer. Given that the average token length for titles is approximately 15.58 and that 99.15% of the titles contain 32 tokens or fewer, the sequence length was fixed at 32. Shorter titles were padded with zeros, and longer titles were truncated to the fixed length.

The tokenization process involved dividing the cleaned titles into subword tokens using WordPiece tokenization. This method splits words into smaller components to handle out-of-vocabulary words effectively. For example, the phrase *Let's do tokenization!* is tokenized into `['Let's', 'do', 'token', '##ization', '!']`. [5]

Each title was transformed into three tensors:

- **Input IDs**: Numerical indices corresponding to each token in the title, including `[CLS]` and `[SEP]`.
- **Attention Mask**: A binary mask indicating the presence of valid tokens (1) and padding (0).
- **Token Type IDs**: Identifies sentence segments, though this was not used for single-sentence titles.

To implement this process, padding was set to `max_length`, truncation was enabled, and the resulting embeddings were returned as PyTorch tensors. By embedding the title data into fixed-length sequences with `[CLS]` and `[SEP]` tokens, we prepared the inputs for BERT's attention mechanisms to capture meaningful contextual relationships within the titles effectively.

D. BERT

In recent years, Transformer-based models have dominated natural language processing (NLP). Among these, BERT [6] is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial taskspecific architecture modifications. In this project, we utilized the pre-trained bert-base-uncased model, which does not differentiate between uppercase and lowercase letters. All input text was converted to lowercase during preprocessing to ensure compatibility with the model's architecture.

1) Model Fine-Tuning: The model was trained for three epochs with the AdamW optimizer. The input data included tokenized titles with fixed sequence lengths of 32, along with attention masks to indicate valid tokens. Loss values steadily decreased across epochs, indicating successful convergence. For example:

- **Epoch 1**: Loss = 0.0361
- **Epoch 3**: Loss = 0.0188

To optimize the model, a grid search was conducted over a set of hyperparameters, including:

- Learning rates (`lr`): {2e-5, 3e-5}
- Epsilon values (`eps`): {1e-8}

IV. Results

The best model was obtained with `lr=2e-5` and `eps=1e-8`, and the results are summarized below. This approach ensured that the fine-tuned model was well-calibrated for the fake news detection task.

TABLE I
PERFORMANCE SCORES

Dataset	Accuracy (%)
Validation Set	92.09
Test Set	92.73

TABLE II
KEY METRICS (TEST SET)

Metric	Overall (%)
Precision	93
Recall	93
F1-Score	93

V. Conclusion

In recent years, fake news detection has become increasingly important in addressing the challenges of misinformation in digital spaces. In this paper, we focused on the implementation of a BERT-based transformer model for fake news detection, using the WELFake dataset. By applying preprocessing steps and sentence embedding techniques, we optimized the input for the model and fine-tuned it for classification tasks.

The experimental results highlight the strong performance of the fine-tuned BERT model, achieving a validation accuracy of 92.09% and a test accuracy of 92.73%. These findings demonstrate the potential of BERT in effectively addressing fake news detection challenges.

References

[1] P. Verma, "The Rise of AI Fake News Is Creating a 'Misinformation Superspreader,"' *Washington Post*, 2023.

[2] S. Shahane, "Fake News Classification," Kaggle, Available: https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification, Accessed: 2024.

[3] R. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake News Detection in Social Media with a BERT-Based Deep Learning Approach," *Multimedia Tools and Applications*, vol. 80, 2021, pp. 11765–11788.

[4] S. Kula, M. Choraś, and R. Kozik, "Application of the BERT-Based Architecture in Fake News Detection," in *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*, Á. Herrero et al., Eds., vol. 1267, Springer, Cham, 2021, pp. 238–247.

[5] Hugging Face, "Tokenization and the WordPiece Tokenizer," Available: https://huggingface.co/learn/nlp-course/en/chapter2/4, Accessed: 2024.

[6] J. Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] OpenAI, "What are tokens and how to count them," Available: https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them, Accessed: 2024.