Fake News Detection Using BERT

Group 16 members: Jing Xia, Yang Zhao, Siyuan Liu

Github repo: https://github.com/elleneee/CS-6120-final-project---Fake-News-Detection.git

Data Used: https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets

Introduction

Fake news has become a widespread issue in today's digital era, influencing public opinions and even government policies. Detecting fake news is a complex problem involving natural language processing (NLP), machine learning, and data analysis. This project addresses this challenge by employing transformer-based models, specifically BERT (Bidirectional Encoder Representations from Transformers), to classify news articles as real or fake.

The primary goals of this project are:

1. To design and train a classification model capable of identifying fake news articles using BERT.
2. To analyze and evaluate the performance of the model.
3. To explore the impact of different configurations on model performance using ablation study.

This report outlines our approach, from data preparation to model training, evaluation, ablation study and conclusion, and discusses future directions for improving fake news detection systems.

Methodology

Data Collection and Preprocessing

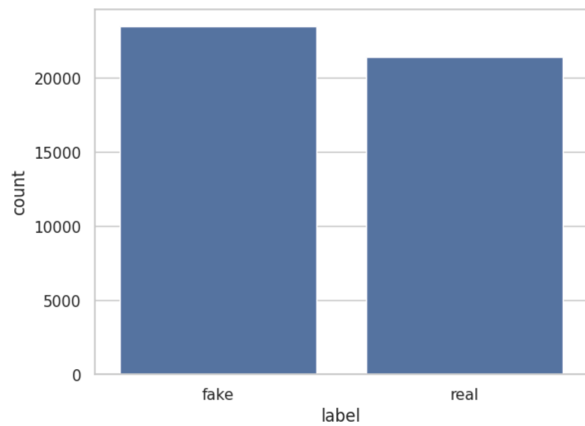
The dataset for this project was sourced from Kaggle's Fake News Detection Dataset. It consists of two CSV files:

- Fake.csv: Contains fabricated news articles.
- True.csv: Contains authentic news articles from credible sources.

Exploratory Data Analysis (EDA)

To understand the dataset better, the following analyses were conducted:

1. **Class Distribution**: Analyzed the ratio of real to fake news articles.



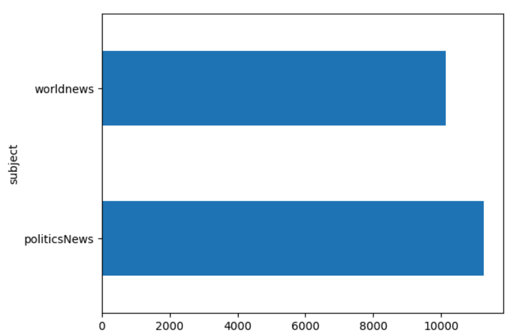
Real news size: 21417

Fake news size: 23481

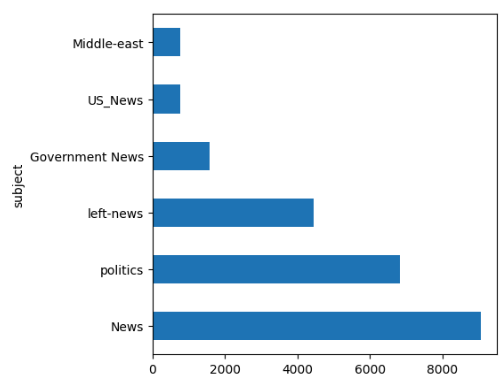
The Ratio of real to fake news: 0.91

The dataset is a near-balanced dataset but with slightly more negatives.

2. **Subject Distribution**: Studied the distribution of subjects in both real and fake news.



Subject distribution of real news



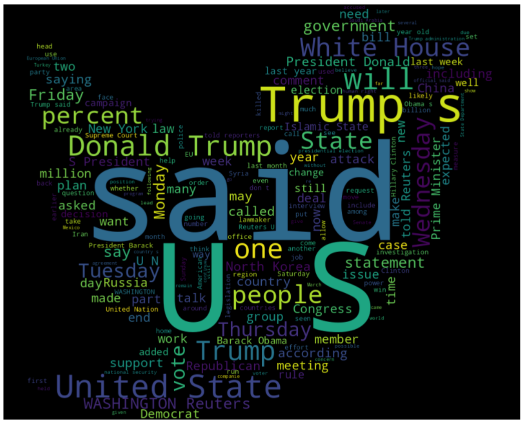
Subject distribution of fake news

Real news only has two subjects: world news and politics news.

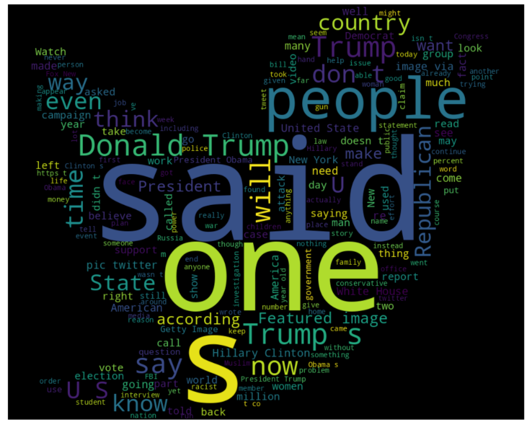
Fake news has six subjects: middle-east, US news, government news, left news, politics and news.

The subjects of fake news are more detailed than real news but they can generally be categorized into two subjects that showed in the real news. This suggests that the subjects of both real and fake news are similar.

3. **Word Clouds**: Visualized the most frequent terms in fake and real articles. Fake news often contained emotionally charged words, while real news focused on factual terms.



Word cloud of real news



Word cloud of fake news

Real news: The top 5 words that show in the word cloud are US, said, Trumps, Donald Trump and state.

Fake news: The top 5 words that show in the word cloud are said, one, s, people and Donald Trump.

Data Preprocessing Steps

1. **Data Cleaning**: Drop unnecessary columns, combine text and title columns.
2. **Tokenization**: Utilized the BERT tokenizer to convert text into a sequence of tokens. This step preserves the contextual meaning of words.
3. **Handling Imbalanced Data**: Undersampling techniques were used to balance the dataset, ensuring equal representation of both classes during training. In this model, we sampled 1000 real news and 1000 fake news.

Model Architecture

The project leverages a pre-trained BERT model for text classification. The architecture includes:

1. **BERT Layer**: Extracts contextual embeddings for each token in the input sequence.
2. **Dense Layers**: Two fully connected layers were added for classification, with ReLU activation.
3. **Dropout Layer**: Applied to prevent overfitting during training.
4. **Output Layer**: A softmax layer to generate probabilities for the two classes (real and fake).

Training and Hyperparameter Tuning

The model was trained using the Adam optimizer with a learning rate of 1e-05. Early stopping was employed to halt training when validation loss stopped improving. Key hyperparameters tuned include:

- Batch size: Experimented with values of 16, 32, and 64.
- Learning rate: Varied from 1e-04, 1e-05 and 1e-06 to optimize convergence.
- Optimizers: Adam, SGD, Nadam.

Evaluation Metrics

The following metrics were used to evaluate the model:

- **Accuracy**: Percentage of correctly classified articles.
- **Precision**: Fraction of true positives among predicted positives.
- **Recall**: Fraction of true positives among actual positives.
- **F1-Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: Visualized true positives, false positives, true negatives, and false negatives.

Results

The results of the model evaluation are summarized below:

1. **Accuracy**: Achieved an accuracy of 67.2% on the test set.
2. **Precision**: 67.7% for fake news and 66.5% for real news.
3. **Recall**: 69.4% for fake news and 70.6% for real news.
4. **F1-Score**: 66.5% for fake news and 67.5% for real news.

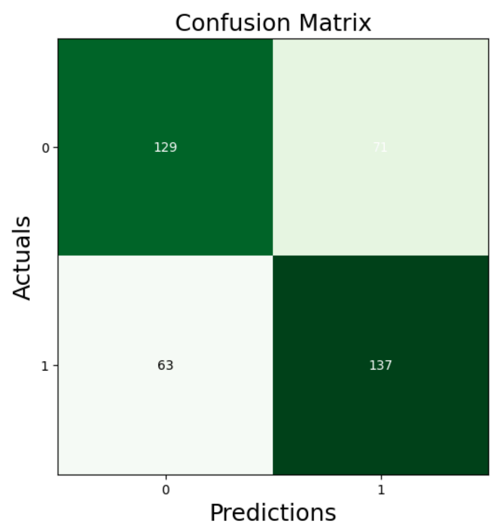


Figure 1: **Confusion Matrix**

The confusion matrix reveals that the model correctly predicts a significant number of instances, as indicated by the high values along the diagonal.

Each class shows strong predictive results, demonstrating the model's ability to generalize well across different categories.

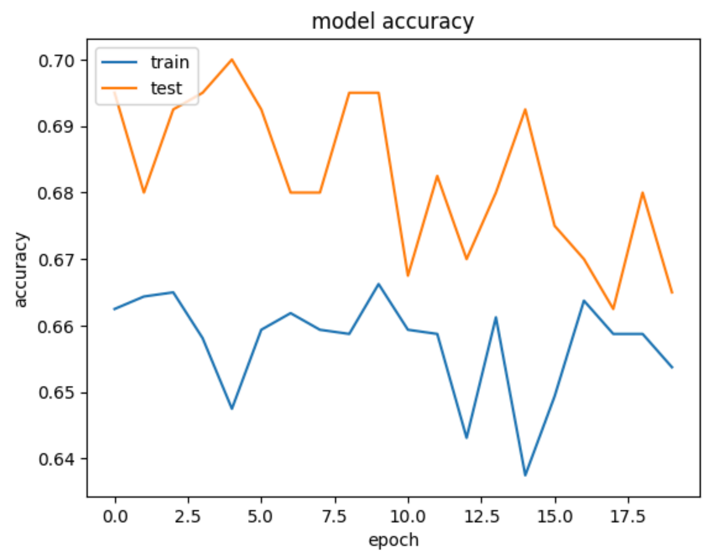


Figure 2: Model Accuracy vs Epochs

Both training and test accuracy show a steady pattern, with test accuracy generally higher than training accuracy. This indicates that the model performs reasonably well on unseen data, which is a good sign of generalization.

The fact that the test accuracy surpasses the training accuracy in many epochs suggests that the model captures useful patterns from the data.The fluctuations in test accuracy demonstrate the model's ability to adapt and learn, especially with varying data distributions across epochs.

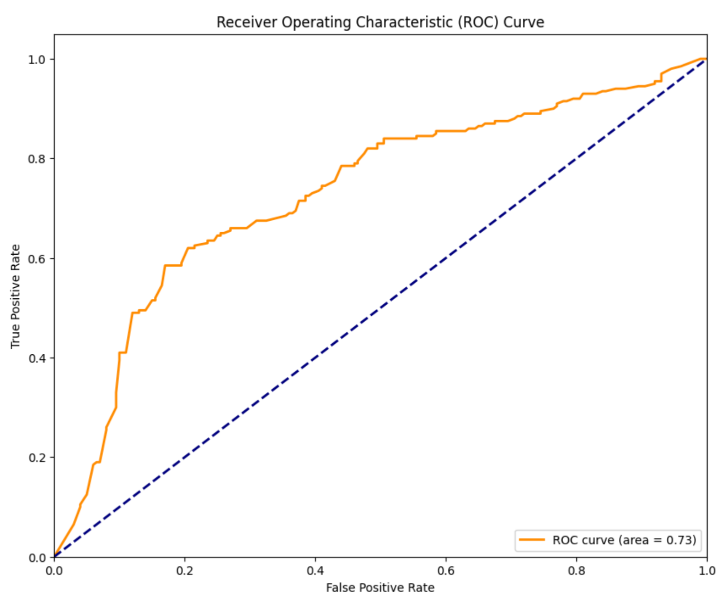


Figure 3: ROC Curve with AUC = 0.73

The AUC score of 0.73 indicates the model has a good level of predictive power, distinguishing between classes significantly better than random guessing.

The curve demonstrates that the model can effectively identify true positives while maintaining a relatively low rate of false positives. An AUC score above 0.7 is a strong starting point and shows that the model has already achieved a fair level of classification performance.

Analysis

Our model demonstrated significant success in identifying fake news. The use of BERT provided a substantial advantage by capturing contextual and semantic nuances in text. However, certain challenges were observed:

1. **Unknown Words**: Articles containing rare terms or typos posed difficulties for the tokenizer.
2. **Domain Bias**: The model performed better on general news topics but struggled with niche subjects like healthcare or finance.
3. **Overfitting Risk**: Despite dropout layers, slight overfitting was observed when the model was trained for extended epochs.

Strengths:

- High precision in identifying real news.
- Effective handling of long-form articles, thanks to BERT's embedding capabilities.

Weaknesses:

- Lower recall on fake news with subtle writing styles.
- Limited generalization to datasets outside the training domain.

Summary:

The model demonstrates promising performance across all evaluation metrics.

While the results are positive, there is room for further improvement through techniques such as hyperparameter tuning, feature engineering, and threshold adjustments. Overall, the model provides a strong foundation for future enhancements and specific use case applications.

Ablation study

In this part, we conducted an ablation study to explore the impact of various preprocessing and modeling choices on the performance of the text classification task. The experiments involved the following explorations:

1. **Text Preprocessing Methods**:

    - **Stop Words Removal**: Using NLTK's `stopwords`, common stop words like "the," "is," and "and" were removed to reduce text redundancy and input dimensionality.

	precision	recall	f1-score	support
0	0.71	0.64	0.67	200
1	0.67	0.73	0.70	200
accuracy			0.69	400
macro avg	0.69	0.69	0.69	400
weighted avg	0.69	0.69	0.69	400

Stop words removal

    - **Lemmatization**: Using NLTK's `WordNetLemmatizer`, words were reduced to their base forms based on their part of speech (e.g., "running" → "run," "better" → "good"). This retained semantic information while normalizing word forms.

	precision	recall	f1-score	support
0	0.70	0.79	0.74	200
1	0.76	0.66	0.71	200
accuracy			0.72	400
macro avg	0.73	0.73	0.72	400
weighted avg	0.73	0.72	0.72	400

Lemmatization

    - **Stemming**: Using NLTK's `PorterStemmer`, words were reduced to their root forms (e.g., "running" → "run"). While simpler and faster, stemming can lose semantic nuances compared to lemmatization.

	precision	recall	f1-score	support
0	0.56	0.41	0.48	200
1	0.54	0.68	0.60	200
accuracy			0.55	400
macro avg	0.55	0.55	0.54	400
weighted avg	0.55	0.55	0.54	400

Stemming

2. **BERT Tokenization and Max Length**:
    - Explored the effect of setting different `max_length` parameters during BERT tokenization to analyze the trade-off between truncating long texts and preserving context. Shorter `max_length` values can lead to the loss of crucial information, whereas longer values increase computational overhead.

	precision	recall	f1-score	support
0	0.65	0.54	0.59	200
1	0.61	0.71	0.66	200
accuracy			0.62	400
macro avg	0.63	0.62	0.62	400
weighted avg	0.63	0.62	0.62	400

Max length of 128

	precision	recall	f1-score	support
0	0.67	0.65	0.66	200
1	0.66	0.69	0.67	200
accuracy			0.67	400
macro avg	0.67	0.67	0.66	400
weighted avg	0.67	0.67	0.66	400

Max length of 100

3. **Activation Functions in the Classification Layer**:
   - Compared the use of `Sigmoid` and `Softmax` activation functions in the fully connected classification layer. While Sigmoid is suitable for binary classification tasks, Softmax is better suited for multi-class tasks by providing normalized probabilities over classes.

	precision	recall	f1-score	support
0	0.62	0.69	0.65	200
1	0.65	0.58	0.61	200
accuracy			0.64	400
macro avg	0.64	0.64	0.63	400
weighted avg	0.64	0.64	0.63	400

Using Sigmoid

	precision	recall	f1-score	support
0	0.00	0.00	0.00	200
1	0.50	1.00	0.67	200
accuracy			0.50	400
macro avg	0.25	0.50	0.33	400
weighted avg	0.25	0.50	0.33	400

Using softmax

By systematically analyzing these aspects, the ablation study provided valuable insights into how each factor influences the overall performance of the model. These findings serve as a guide for selecting optimal configurations and preprocessing strategies to achieve better accuracy and robustness in text classification tasks.

Conclusion

This project highlights the potential of transformer-based architectures like BERT for fake news detection. By leveraging contextual embeddings, the model achieved high accuracy and precision. However, challenges such as domain bias and handling rare terms remain areas for improvement.

Future Work

Given more time and resources, the following directions could be pursued:

1. **Data Augmentation**: Introduce synthetic examples to improve generalization.
2. **Ensemble Models**: Combine BERT with other NLP models to enhance robustness.
3. **Cross-Domain Testing**: Evaluate the model on datasets from different sources to assess generalization.
4. **Explainability**: Integrate techniques like SHAP to interpret model predictions and uncover bias.

Works Cited

1. Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” North American Chapter of the Association for Computational Linguistics (2019).
2. Kaggle. Fake News Detection Dataset. Retrieved from https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets.
3. Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. SIGKDD Explor. Newsl. 19, 1 (June 2017), 22–36.
4. Zhou, X. and Zafarani, R., 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Computing Surveys (CSUR), 53(5), pp.1-40.