Fake News Detection with Logistic Regression and Advanced CNN

Kushal Bhattarai
kushal.bhattarai@student.lut.fi

Taha Ahmed
taha.ahmed@student.lut.fi

April 20, 2025

1 Data Processing

We preprocessed the dataset consisting of news articles to detect fake news. As the full dataset size was approximately 30GB, we used around 10% of it due to hardware constraints. The dataset was split into training (80%), validation (10%), and test (10%) sets. We selected relevant columns: Domain, Title, Authors, Type, Content, and URL.

The preprocessing steps included:

- Converting text to lowercase
- Removing digits and special characters
- Tokenizing text using NLTK's `word_tokenize`
- Removing stopwords
- POS tagging and lemmatizing tokens using WordNetLemmatizer

Lemmatization was chosen over stemming for producing dictionary-formed root words [1]. For each article, a binary label was assigned: '1' for reliable news (type = 'reliable') and '0' for other types like fake, satire, hate, etc. The text content was then transformed into TF-IDF features.

To better understand the processed data:

- **Figure 2** shows the class distribution. The dataset is imbalanced, with more fake news samples than not fake, which could influence the model's predictions.
- **Figure 3** displays the top 20 most frequent words in the dataset. Terms like *"say," "trump," "blockchain," "state"* are among the most common, suggesting topic trends.
- **Figure 1** presents the distribution of text lengths. Most articles are under 1000 words, but some are very lengthy. This distribution validates the choice of using a traditional model like logistic regression.

2 Model: Logistic regression

We used a logistic regression classifier with class weighting and a TF-IDF vectorized representation of the cleaned text. The model was trained with the following configuration [3]:

- `TfidfVectorizer(min_df=2, max_df=0.8)`
- `LogisticRegression(class_weight='balanced', max_iter=1000)`

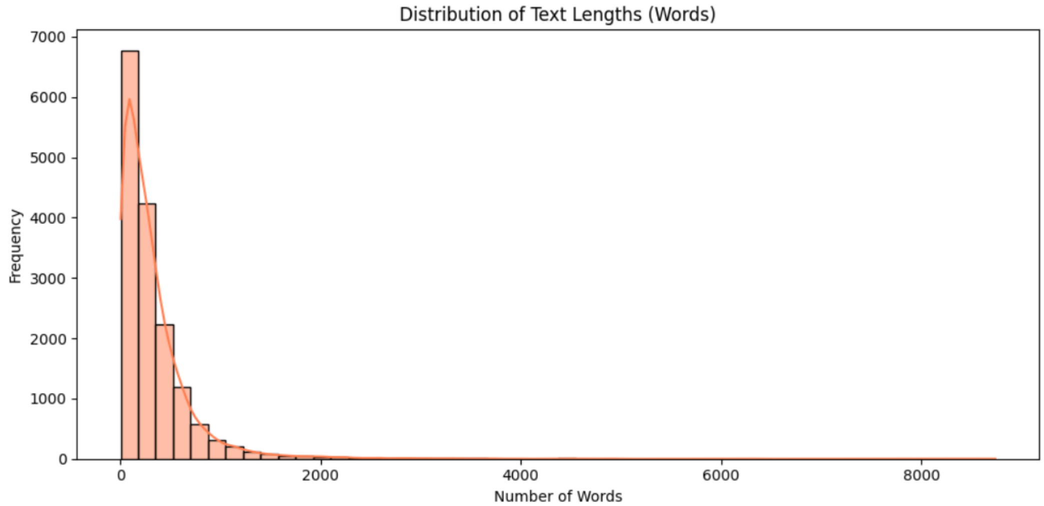

Figure 1: Distribution of Text Lengths (Words)

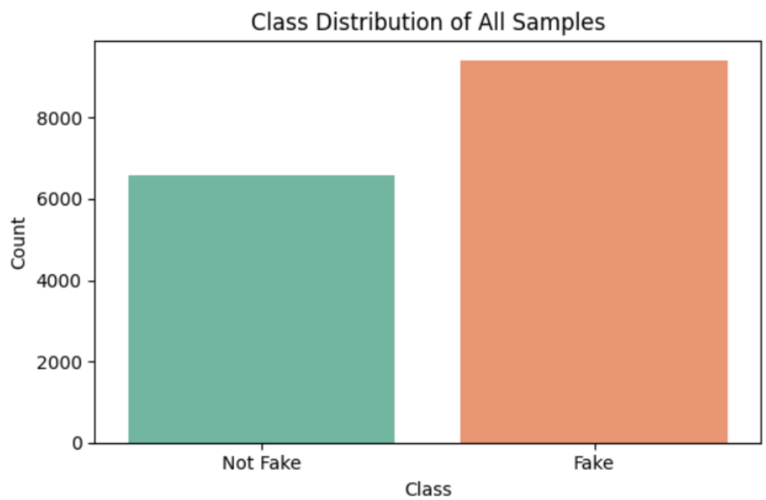

Figure 2: Class Distribution of All Samples

Final Evaluation

Validation Accuracy: `0.8857`
Test Accuracy: `0.9023`

Validation Report

```
              precision    recall  f1-score   support

    Not Fake       0.83      0.92      0.87      1293
        Fake       0.93      0.86      0.90      1707

    accuracy                           0.89      3000
   macro avg       0.88      0.89      0.88      3000
```

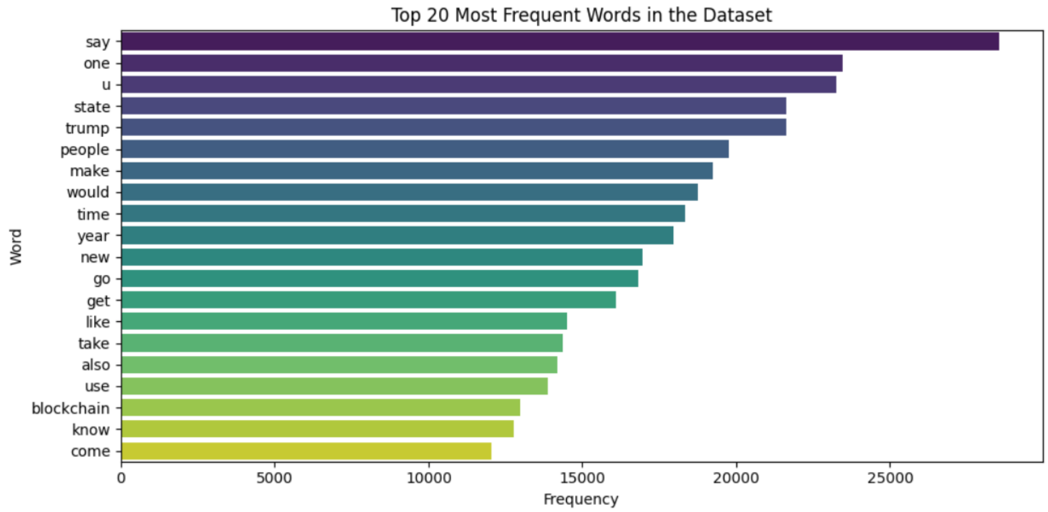

Figure 3: Top 20 Most Frequent Words in the Dataset

weighted avg	0.89	0.89	0.89	3000

Test Report

	precision	recall	f1-score	support
Not Fake	0.84	0.94	0.89	1235
Fake	0.95	0.88	0.91	1765
accuracy			0.90	3000
macro avg	0.90	0.91	0.90	3000
weighted avg	0.91	0.90	0.90	3000

Cross-Dataset Generalization: LIAR Dataset Evaluation

To test model robustness, we evaluated the logistic regression model on the LIAR dataset. Results indicate poor generalization, especially for detecting fake news in different formats.

Validation Accuracy on LIAR: 0.5140
Test Accuracy on LIAR: 0.5556

LIAR Validation Report

	precision	recall	f1-score	support
Not Fake	0.52	0.93	0.67	668
Fake	0.45	0.06	0.11	616
accuracy			0.51	1284
macro avg	0.48	0.50	0.39	1284
weighted avg	0.49	0.51	0.40	1284

LIAR Test Report

	precision	recall	f1-score	support
Not Fake	0.56	0.93	0.70	714
Fake	0.44	0.07	0.12	553
accuracy			0.56	1267
macro avg	0.50	0.50	0.41	1267
weighted avg	0.51	0.56	0.45	1267

3 Advanced Model: Enhanced CNN with Attention

To improve model generalization and leverage advanced architectures, we implemented a multiscale convolutional neural network (CNN) with attention mechanisms [2].

Architecture Overview

The model applies three parallel 1D convolution layers with different kernel sizes (3, 5, 7) to capture short and long-term dependencies in the TF-IDF feature space. Each convolution output is batch normalized, passed through a ReLU activation, and modulated via an attention mechanism to focus on informative features. The output of the three paths is concatenated and passed through fully connected layers for classification.

- Input: TF-IDF vectors (max 10,000 features)
- Layers: Conv1D (3, 5, 7) → BatchNorm → Attention → FC Layers
- Optimizer: AdamW with learning rate scheduling
- Loss: CrossEntropy with class weights
- Strategy: WeightedRandomSampler to address imbalance

Performance Summary

Validation Accuracy: `0.8638`
Test Accuracy: `0.8562`

Validation Report

	precision	recall	f1-score	support
Not Fake	0.75	1.00	0.86	2063
Fake	1.00	0.77	0.87	2937
Accuracy			0.86	5000
Macro avg	0.88	0.88	0.86	5000
Weighted avg	0.90	0.86	0.86	5000

Test Report

	precision	recall	f1-score	support
Not Fake	0.75	0.99	0.85	2087
Fake	0.99	0.76	0.86	2913
Accuracy			0.86	5000
Macro avg	0.87	0.88	0.86	5000
Weighted avg	0.89	0.86	0.86	5000

Cross-Dataset Generalization: LIAR Dataset Evaluation

To evaluate the generalizability of the CNN model, we tested it on the LIAR dataset. While the model achieved decent accuracy, performance dropped compared to the original dataset, especially in detecting fake news.

LIAR Validation Accuracy: `0.5179`

LIAR Validation Report

	precision	recall	f1-score	support
Not Fake	0.52	0.95	0.67	668
Fake	0.47	0.05	0.08	616
accuracy			0.52	1284
macro avg	0.50	0.50	0.38	1284
weighted avg	0.50	0.52	0.39	1284

LIAR Test Accuracy: `0.5509`

LIAR Test Report

	precision	recall	f1-score	support
Not Fake	0.56	0.95	0.71	714
Fake	0.35	0.03	0.06	553
accuracy			0.55	1267
macro avg	0.45	0.49	0.38	1267
weighted avg	0.47	0.55	0.42	1267

Training Analysis

Figure 4 shows the loss of training and the precision of the validation over epochs. The model quickly reduces loss and achieves high validation accuracy, although there is some fluctuation that indicates sensitivity to learning rate and possible overfitting in early epochs.

4 Conclusion

The logistic regression model performed exceptionally well on the FakeNewsCorpus dataset, achieving a validation accuracy of 88.57% and a test accuracy of 90.23%. The classification report shows

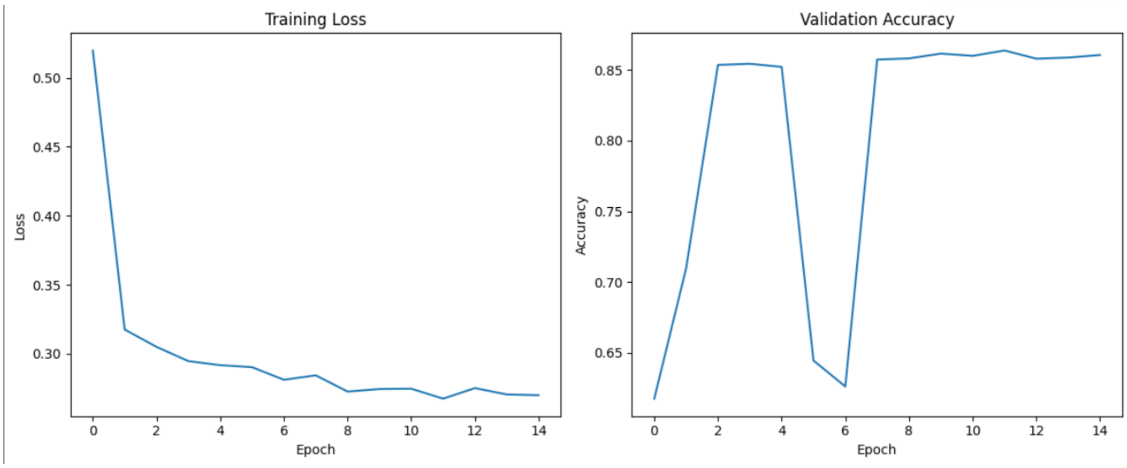

Figure 4: Training Loss and Validation Accuracy of the Enhanced CNN Model

high precision and recall for both fake and not fake classes, indicating balanced learning. However, the model struggled to generalize when tested on the LIAR dataset. Despite maintaining a high precision for the ‘Not Fake‘ class, it exhibited extremely low recall for the ‘Fake‘ class, highlighting poor adaptability to different data formats and writing styles.

In contrast, the enhanced CNN model with attention mechanisms demonstrated more robust generalization. While slightly underperforming compared to logistic regression on the in-domain test set (85.62% accuracy), it maintained a better balance in precision and recall across both classes, especially in handling class imbalance and capturing semantic nuances. The use of multi-scale convolutions and attention contributed to its capability to extract deeper contextual features from the TF-IDF inputs.

These findings emphasize the trade-off between simplicity and generalizability. Logistic regression, though fast and interpretable, is sensitive to the domain it is trained on. Meanwhile, deep models like CNNs, albeit computationally expensive, offer greater resilience across diverse datasets when properly regularized. Future work could explore domain adaptation techniques and hybrid architectures to further improve cross-dataset generalization.

References

[1] Murel J. and Kavlakoglu E. *What are stemming and lemmatization?* IBM, 2023. https://www.ibm.com/think/topics/stemming-lemmatization

[2] GeeksforGeeks. *Text Classification using CNN*, 2023. https://www.geeksforgeeks.org/text-classification-using-cnn/

[3] GeeksforGeeks. *Text Classification using Logistic Regression*, 2023. https://www.geeksforgeeks.org/text-classification-using-logistic-regression/