Classifying Fake News on the WELFake dataset

Vinicius Eller
Computer Engineering
Insper
São Paulo, Brazil
viniciusge@al.insper.edu.br

I. Dataset

This project utilizes the WELFake dataset [1], which combines data from various sources, including Kaggle, McIntire, Reuters, and BuzzFeed Political, in order to mitigate overfitting. The dataset business application is relevant to platforms that combat misinformation, such as social media networks and news agencies.

It contains a total of 72,134 articles, with 35,028 labeled as reliable news (R-News) and 37,106 labeled as fake news (F-News). To focus purely on the content, only the ”label” and ”text” columns are considered for this analysis, discarding other features like URLs or titles that could introduce bias. This ensures that the analysis emphasizes the actual content of the news articles, aligning with the main objective of detecting fake news based on linguistic characteristics rather than external features.

II. Classification pipeline

The classification pipeline implemented in this project, while inspired by the WELFake framework, does not aim to achieve state-of-the-art performance in fake news detection. Instead, it employs a foundational approach

The pipeline begins with data cleaning, removing articles with titles only, which reduced the dataset to 37,067 F-News articles, while R-news remained unchanged. Subsequently, English stop words are removed using the Natural Language Toolkit (NLTK) library [2].

Data processing is performed using NLTK in conjunction with WordNet [3], allowing for the lemmatization of the dataset.

The text data was then transformed into a matrix of token counts using the CountVectorizer from scikit-learn [5], where only lowercase terms were considered to avoid case sensitivity. Terms with frequencies that were too high or too low across the entire dataset were removed to reduce noise. Specifically, terms that appeared in fewer than 5% or more than 80% of the articles were excluded based on experimentation.

Instead of using raw token counts, binary features were employed, indicating whether a term was present or absent in an article. This approach was chosen because news may frequently repeat specific terms, introducing bias that could mislead the classifier.

The classification model selected was Logistic Regression, implemented using scikit-learn with default parameters, including L2 regularization and the Limited-memory BFGS (LBFGS) solver.

III. Evaluation

The model was evaluated over 100 different splits of the dataset, maintaining an 80/20 ratio for training and testing data, respectively. The proportion of F-News and R-News was preserved across these splits to ensure balanced evaluation. The results are detailed in Table I.

Logistic Regression’s interpretability makes it possible to analyze the most influential words in classification based on the learned coefficients. For example, Figure 2 highlights the most significant terms for R-News classification. A bias is evident, as the dataset contains reliable news articles primarily sourced from Reuters [6], a British news agency.

A similar bias is observed when analyzing the important features for F-News classification in Figure 1. Terms such as ”getty,” ”featured,” and ”image” are significant predictors of F-News. This is due to the presence of fake news articles referencing Getty Images [7], an image database.

IV. Dataset size

Examining the error rates on both the training and testing datasets across varying levels of downsampling. Figure 3 shows the convergence of the train and test error rates as the number of samples increases. This indicates that merely adding more data is unlikely to yield improvements in accuracy.

Fake news detection is constantly evolving. One viable approach to improve model generalization would be to include data that is currently underrepresented in the dataset, such as articles from more recent years or from sources outside the scope of the 2016 U.S. elections, which dominate this dataset.

For business applications, diversifying the data (e.g., adding social media posts or international news) would enhance the model’s adaptability across different contexts, rather than merely increasing the volume of similar data.

V. Topic analysis

Topic modeling was performed using Non-Negative Matrix Factorization (NMF) decomposition from scikit-learn, which allowed the dataset to be divided into two primary topics: *Generic News* and *U.S. Election News*.

Following the same classification pipeline outlined in Section II, Logistic Regression models were trained separately for each topic. The evaluation scores for these models are presented in Table II.

VI. Appendix

TABLE I
Score Evaluation

Accuracy	**Precision**	**Recall**	**F1**
0.9289	0.9295	0.9293	0.9293

TABLE II
Score Evaluation: *Generic Topic*

Topic	**Accuracy**	**Precision**	**Recall**	**F1**
Generic	0.9284	0.9356	0.9354	0.9355
Elections	0.8860	0.9220	0.9231	0.9222

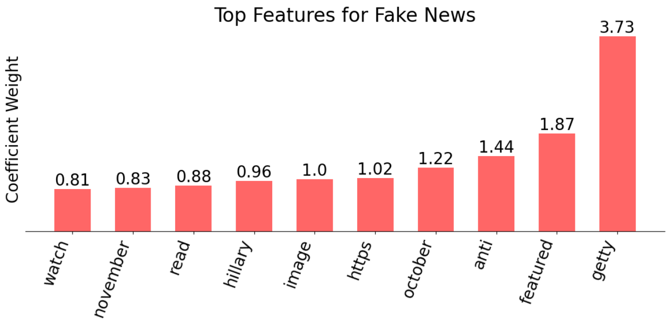

Fig. 1. Word importance for model classification (Fake News).

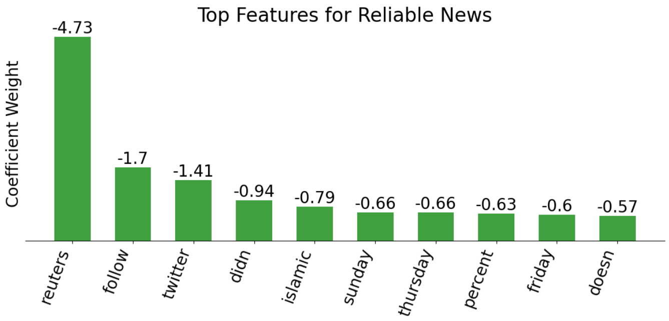

Fig. 2. Word importance for model classification (Reliable News).

References

[1] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, "WELFake: Word Embedding Over Linguistic Features for Fake News Detection," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 881-893, Aug. 2021, doi: 10.1109/TCSS.2021.3068519.

[2] NLTK, "Natural Language Toolkit," [Online]. Available: https://www.nltk.org/. [Accessed: Oct. 3, 2024].

[3] WordNet, "A lexical database for the English language," [Online]. Available: https://wordnet.princeton.edu/. [Accessed: Oct. 3, 2024].

[4] C. Fellbaum, "WordNet and wordnets," in *Encyclopedia of Language and Linguistics*, 2nd ed., K. Brown, Ed., Oxford: Elsevier, 2005, pp. 665-670.

[5] scikit-learn, "sklearn," [Online]. Available: https://scikit-learn.org/ [Accessed: Oct. 3, 2024].

[6] Reuters, "Reuters," [Online]. Available: https://www.reuters.com/. [Accessed: Oct. 3, 2024].

[7] Getty Images, "Getty Images," [Online]. Available: https://www.gettyimages.com.br/. [Accessed: Oct. 3, 2024].

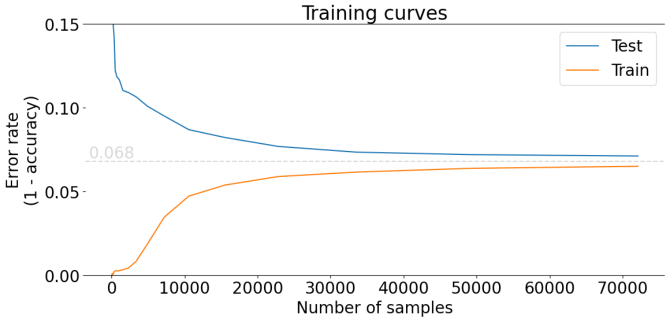

Fig. 3. Error rate versus number of samples training.