Fake News Detection Using DeBERTa-v3 and SHAP

By: Rustam Aslam and Shubham Jagtap

1. Introduction

The advent of digital media has transformed how information is consumed. While this transformation has facilitated rapid information sharing, it has also led to the widespread circulation of misinformation, more commonly known as "fake news." The societal impact of fake news has been significant, influencing public opinion, policy-making, and even election outcomes. Consequently, developing systems that can automatically detect fake news with high accuracy and offer interpretable results has become a critical area of research in natural language processing (NLP) and machine learning (ML).

This project focuses on solving two major challenges in fake news detection: making accurate predictions and helping people understand why a decision was made. To do this, it uses DeBERTa-v3, a strong language model developed by Microsoft. DeBERTa-v3 is based on the transformer architecture and is known for its good performance on tasks that involve understanding language. In this project, the model is used to tell whether a news article is real or fake, using only the article's text.

However, it's not enough for a model to just make the right prediction. In many cases, it's just as important to understand why a model made a certain choice, especially when the decisions can affect public trust. That's why this project also includes SHAP (SHapley Additive exPlanations), a tool that helps explain model predictions in a clear way. SHAP works by showing which words or parts of a news article were most important in the model's decision. This makes it easier for users to trust the system and for developers to see how the model works internally.

By combining a powerful language model with tools for explanation, this project aims to build a fake news detection system that is both accurate and transparent. It can help people check whether a news story is real or fake, and also understand the reasoning behind the answer.

2. Methodology

The development process was divided into several stages, including dataset preparation, model selection and training, and explainability integration. Each stage was designed with a focus on balancing model performance and interpretability.

2.1 Dataset and Preprocessing

The dataset utilized in this project was the “FakeNewsDetection” dataset available on Kaggle. This dataset contains thousands of labeled news articles, each annotated with a binary label indicating whether the content is real (0) or fake (1). From this point forward, we refer to these as reliable (0) and unreliable (1) for clarity. The dataset includes the textual content of the articles, along with corresponding metadata such as titles and sources. For the purpose of this project, only the main article text and the associated labels were retained to focus exclusively on content-based classification.

The articles span a diverse range of topics and writing styles, reflecting both credible journalism and misleading or fabricated content. This diversity introduces real-world complexity to the task, such as variations in tone, structure, and use of persuasive language, challenges that a robust model must learn to handle.

Before training the model, a series of preprocessing steps were applied to clean and standardize the text data:

- Column Selection: Only the relevant columns, specifically the article text and its corresponding label, were retained.
- Handling Missing Data: Any entries with missing values were removed to ensure data quality and prevent inconsistencies during model training.
- Text Cleaning:
    - All characters were converted to lowercase to avoid discrepancies due to case sensitivity.
    - Extra white space was removed for formatting consistency.
    - Non-alphanumeric characters were stripped out (except basic punctuation such as periods and commas) using regular expressions. This helped reduce textual noise while preserving sentence structure.

These preprocessing steps ensured that the dataset was in a clean and consistent format, making it suitable for tokenization and model input. Proper text preparation played a crucial role in stabilizing the training process and improving the performance of the DeBERTa-v3 model used in this project.

2.2 Model Selection and Training Approach

To build an effective text classification system for fake news detection, this project employed the DeBERTa-v3 model developed by Microsoft. DeBERTa-v3 is an improved version of the BERT architecture, incorporating enhancements like a smarter way to focus on important words and understand their order in a sentence. These features allow the model to more effectively capture the meaning and context of language, making it well-suited for tasks that require deep

understanding of text. Its advanced architecture makes it a strong choice for accurately identifying whether a news article is reliable or unreliable.

Once the model and its associated tokenizer were loaded, the text data was preprocessed and passed through the tokenizer to convert it into a format suitable for model input. Each article was padded or truncated to a fixed length (512 tokens) to ensure consistent input size, a critical step for transformer-based models like DeBERTa.

The dataset was then split into training and testing sets using an 80-20 split. A stratified sampling approach was employed to preserve the ratio of reliable and unreliable news articles across both sets, ensuring balanced class representation during training and evaluation.

To train the model, Hugging Face's Trainer API was used. This API simplifies the training process by handling the complexities of model optimization, evaluation, and checkpointing behind the scenes. However, it still allows for detailed customization. The training configuration included:

- 15 training epochs, with early stopping triggered after 3 non-improving evaluation rounds to prevent overfitting.
- A batch size of 8 per device for both training and evaluation phases.
- A learning rate of 2e-5, with a linear decay schedule and 500 warmup steps to stabilize early learning.
- F1-score was selected as the main metric to monitor, since it balances precision and recall, especially important in binary classification problems.

To evaluate the model's performance during training, a custom metrics function was defined. This function calculated accuracy, precision, recall, and F1-score using the predictions and true labels from the evaluation set. The model that performed best based on F1-score was automatically selected and saved at the end of training.

2.3 Model Explainability with SHAP

To enhance the interpretability of the model's predictions, SHAP (SHapley Additive exPlanations) was integrated into the pipeline. SHAP provides a game-theoretic approach to explain the output of machine learning models by attributing contributions to individual input features, in this case, words in a news article. The implementation began by wrapping the trained DeBERTa-v3 model and tokenizer into a Hugging Face pipeline, which served as the prediction function for SHAP. A Text masker was employed to mask tokens while preserving sentence structure, enabling more realistic changes to the input for explanation. The SHAP explainer was then used to analyze three types of samples from the test set: a correctly predicted reliable article, a correctly predicted unreliable article, and a misclassified article. For each example, the model's predictions were explained using 'shap.plots.text', which visualizes word-level contributions,

highlighting words that pushed the model towards or away from a particular class. This analysis helped uncover which textual patterns and phrases the model relied upon, thereby increasing the transparency and trustworthiness of the system, and providing valuable insights into both correct decisions and failure cases.

3. Experiments

3.1 Model Performance

Upon completion of the training process, the DeBERTa-v3 model exhibited exceptional performance on the test dataset. The final test accuracy reached 99.61%, while the overall F1-score was 0.9961, indicating the model's high effectiveness in distinguishing between reliable and unreliable news articles. These metrics reflect a strong generalization capability, as the model was able to maintain near-perfect classification performance on unseen data.

A closer look at the evaluation metrics further reinforces the model's reliability:

- Reliable News (Class 0):
    - Precision: 0.9952
    - Recall: 0.9971
    - F1-score: 0.9962
- Unreliable News (Class 1):
    - Precision: 0.9971
    - Recall: 0.9952
    - F1-score: 0.9961

To monitor training dynamics, learning curves for both accuracy and loss were plotted over the course of training epochs. The validation accuracy improved consistently, mirroring trends in the training data, while the validation and training losses gradually declined. This behavior is indicative of a well-generalized model learning meaningful patterns, rather than simply memorizing training samples. The learning curves suggest that the model is neither overfitting nor underfitting. Both training and validation losses remained low and stable across epochs, without significant divergence. Validation accuracy remained consistently high, and the model achieved strong performance on the test set with balanced precision and recall for both classes. These results confirm that the model has learned relevant features from the data and is capable of generalizing effectively to new, unseen examples.

3.2 Explainability Analysis

To better understand how the model makes decisions, SHAP visualizations were used to analyze three different types of predictions: a correctly predicted reliable article, a correctly predicted

unreliable article, and a misclassified case. These examples helped uncover which parts of the text the model considered most important when making its prediction.

For the correctly predicted reliable article, SHAP highlighted neutral and factual language as the key contributors to the prediction. Words and phrases such as “investigating,” “public records,” and “federal law enforcement” had a strong positive influence on the model’s decision to label the article as reliable. This suggests that the model learned to associate objective and formal language with trustworthy content.

In the case of the correctly predicted unreliable article, SHAP revealed that the model focused on emotionally charged and conspiratorial terms like “atrocious torture,” “big pharma,” and “secret eugenics.” These words played a major role in pushing the prediction toward the unreliable class. This shows the model’s ability to detect misleading content by picking up on exaggerated or manipulative language.

Finally, the misclassified example involved an article that was labeled as unreliable in the dataset but was incorrectly predicted as reliable by the model. SHAP showed that although the article contained strong language like “fraud,” “theft,” and “scandal,” the model placed more weight on the structured format and overlooked the critical tone. This type of error highlights an important limitation — the model may sometimes favor grammatical or stylistic features over the underlying intent of the content.

Overall, these SHAP visualizations made the model’s behavior more transparent and helped explain both successful predictions and failure cases. This kind of interpretability is essential for building trust in fake news detection systems and improving their performance over time.

4. Conclusion

This project successfully implemented an end-to-end fake news detection system that combines state-of-the-art NLP techniques with interpretable AI methods. By leveraging the capabilities of DeBERTa-v3 for classification and SHAP for explanation, the system not only achieves high predictive accuracy but also offers transparency in its decision-making process.

In the future, this work can be expanded by incorporating additional datasets, exploring multi-class classification (e.g., satire vs. misinformation vs. propaganda), and experimenting with other explainability techniques such as LIME or attention-based methods. The ultimate goal is to build a comprehensive and trustworthy tool that helps individuals and organizations make informed decisions in today’s fast-moving and information-rich world.