Ziyi Liu, Hanzhe Zhou, Yide Fang

CS 424 Machine Learning

Professor Christian Lopez

05/13/2025

Executive report for Fake News Project

Problem Overview

In an era dominated by rapid information exchange, the spread of fake news has become a critical concern. Misinformation can have far-reaching consequences, influencing elections, damaging reputations, and eroding public trust. The rise of generative AI has further complicated this issue by enabling the creation of highly realistic yet entirely fabricated content. Detecting fake news is no longer just about fact-checking; it requires understanding the subtle linguistic and structural differences between truthful and deceptive narratives.

This project aims to explore the dual challenge of detecting fake news and understanding how such news can be algorithmically generated. We built a complete machine learning pipeline that takes real news data, uses large language models to generate fake counterparts, and trains a classifier to distinguish between the two. The goal is both practical and investigative: build a tool that can identify fakes while analyzing what makes fake news structurally distinct from real reporting.

Data Collection and Generation

We began with the widely used open-source dataset from Kaggle titled "Fake and Real News," which contains approximately 21,000 labeled news articles split between fake and real sources. From this dataset, we extracted 15,000 verified real news articles for use in our study. Each article includes a title, body text, publication date, and subject category.

To create synthetic fake news, we used OpenAI's GPT-3.5-turbo model via its API. For each real article, we supplied a structured prompt asking the model to generate a plausible but entirely false article written in the same style and tone. This resulted in a matched dataset of 15,000 real and 15,000 fake articles. We used careful prompt engineering to ensure that the fake articles preserved journalistic structure while altering the factual content. Each generated article was stored alongside its source article for traceability and evaluation.

Model Architecture

For the classification task, we adopted a traditional machine learning pipeline focused on interpretability and speed. The process includes the following steps:

- Text Cleaning: We used a custom Python function to lowercase all text, remove URLs, special characters, and excess whitespace.
- Vectorization: The cleaned text was vectorized using the TfidfVectorizer from scikit-learn, with a max feature size of 10,000 and bi-gram support to capture phrase structure.
- Classifier: We trained a Logistic Regression model with a maximum of 1000 iterations. The model was trained on 80% of the data and tested on the remaining 20%.

The model and vectorizer were serialized using Python's pickle module for later use in a prediction script. The design decision to use Logistic Regression was driven by its interpretability and its strong performance on binary text classification problems.

Evaluation

Model performance was evaluated using the following metrics:

- **Accuracy**: 80%
- **Confusion Matrix**: Showed near-perfect classification with minimal misclassification (e.g., 1000 misclassified instances in a test set of 6000)

These results suggest that the classifier learned highly reliable patterns distinguishing GPT-generated content from real articles. While the scores are impressive, they may be inflated due to stylistic differences introduced during text generation. As such, while our model is effective for detecting synthetically generated fakes, its generalization to human-written fake news remains an open question.

Conclusions and Future Work

This project successfully demonstrates the feasibility of building a reliable fake news detector using a simple logistic regression model and synthetically generated training data. The model's excellent performance provides a strong foundation for further exploration.

However, there are limitations. The generated fake news was created using a consistent prompt template, which may make it easier for the model to detect compared to real-world fakes. Additionally, the lack of domain variation in both real and synthetic content could limit generalizability.

To address these challenges, future work could explore:

- Using multiple LLMs and varied prompting strategies to create more stylistically diverse fake samples
- Training and comparing deep learning models such as fine-tuned DistilBERT or RoBERTa
- Testing the model on real-world misinformation datasets (e.g., COVID-19 misinformation, political hoaxes)
- Deploying the model as a browser plugin or news verification API
- Creating a feedback loop where detected fakes are reviewed and used to retrain the model over time

This project provides both a working technical solution and a platform for investigating the deeper social and technical aspects of misinformation in the AI era.