Technical Report: Final Project DS 5220: Supervised Machine Learning and Learning Theory

Team Members:
Cameron Jester & Rimsha Kayastha
Khoury College of Computer Sciences
Data Science Program
`jester.c@northeastern.edu, kayastha.r@northeastern.edu`

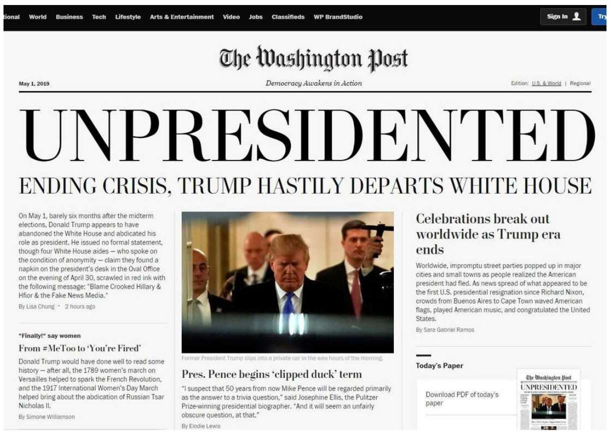
December 9, 2024

Contents

1 Abstract

In the ever-evolving world of media consumption, news outlets and the nature of news itself have undergone immense changes. With tools such as ChatGPT and the new marketing technique of ”clickbait,” the rise in fake news has been dramatic. Fake news infiltrating media platforms can lead to issues such as distrust in news, the inability to make informed decisions, and even social polarization. The motivation for this project is to create a classification model to detect fake news, addressing these challenges.

As 2024 is an election year, ensuring that U.S. citizens are not misled by misinformation is crucial. Candidates may exploit platforms to spread false information about their opponents to gain popularity. A news classification tool that accurately identifies fake news can help mitigate such behavior, fostering trust and informed decision-making in the political landscape and beyond.


ional World Business Tech Lifestyle Arts & Entertainment Video Jobs Classifieds WP BrandStudio
Sign In
Try

The Washington Post

May 1, 2019 *Democracy Awakens in Action* Edition: U.S. & World | Regional

UNPRESIDENTED

ENDING CRISIS, TRUMP HASTILY DEPARTS WHITE HOUSE

On May 1, barely six months after the midterm elections, Donald Trump appears to have abandoned the White House and abdicated his role as president. He issued no formal statement, though four White House aides — who spoke on the condition of anonymity — claim they found a napkin on the president's desk in the Oval Office on the evening of April 30, scrawled in red ink with the following message: "Blame Crooked Hillary & Hfior & the Fake News Media."

By Lisa Chung · 2 hours ago

"Finally!" say women

From #MeToo to 'You're Fired'

Donald Trump would have done well to read some history — after all, the 1789 women's march on Versailles helped to spark the French Revolution, and the 1917 International Women's Day March helped bring about the abdication of Russian Tsar Nicholas II.

By Simone Williamson

Former President Trump slips into a private car in the wee hours of the morning.

Pres. Pence begins 'clipped duck' term

"I suspect that 50 years from now Mike Pence will be regarded primarily as the answer to a trivia question," said Josephine Ellis, the Pulitzer Prize-winning presidential biographer. "And it will seem an unfairly obscure question, at that."

By Elodie Lewis

Celebrations break out worldwide as Trump era ends

Worldwide, impromptu street parties popped up in major cities and small towns as people realized the American president had fled. As news spread of what appeared to be the first U.S. presidential resignation since Richard Nixon, crowds from Buenos Aires to Cape Town waved American flags, played American music, and congratulated the United States.

By Sara Gabriel Ramos

Today's Paper

Download PDF of today's paper

Figure 1: Counterfeit Article Distributed by Activists (Source: NPR)

This supervised machine learning project focuses on text-based data, making it a robust Natural Language Processing (NLP) problem. Due to the inherent complexity of news articles, which often contain nuanced meanings in long phrases, it is essential to choose a model with self-attention mechanisms. For this reason, BERT (Bidirectional Encoder Representations from Transformers) was selected for encoding the text. After encoding with BERT, two classification models—logistic regression and K-Nearest Neighbors (KNN)—were implemented to classify news as real or fake, given their suitability for binary classification.

The dataset for this project was sourced from Kaggle. While other users on Kaggle have utilized BERT for similar tasks, this project adopted the pre-trained HuggingFace architecture for efficiency and fine-tuned it for the specific task. Additionally, the classification models (logistic regression and KNN) provided a novel perspective compared to other approaches found on the platform.

Limitations of this project primarily revolved around time constraints, as only 42 days were available between the start and final presentation, including breaks for holidays. Computational resources also posed significant challenges. One group member's laptop was unable to handle BERT's 340 million parameters locally, and Google Colab's memory limits frequently caused interruptions. These constraints hindered synchronous collaboration and the depth of exploration desired for this project.

2 Experiment Setup

This project utilized the "Title" column of the dataset to train the models instead of the "Text" column, due to BERT's 512-token limitation. Titles provided sufficient information for classification without truncating the main text of articles, ensuring a lossless approach.

2.1 Dataset Statistics

The dataset, sourced from Kaggle, received a usability score of 10.0, indicating its completeness, credibility, and compatibility. It contains 6,256 news articles with four columns: "number," "title," "text," and "label." The "number" column is a unique identifier and was excluded from the analysis. The "title" column contains article titles, while the "text" column contains the article body. The "label" column assigns binary classes, with "real" for genuine articles and "fake" for fabricated ones. Titles and texts include punctuation such as periods, commas, and exclamation points, which were retained during preprocessing.

2.2 Implementation Techniques

The project was implemented using Visual Studio Code (VS Code) as the Integrated Development Environment (IDE). The following essential libraries and frameworks were installed and imported:

```python
import pandas as pd
from bs4 import BeautifulSoup
import re
import torch
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from transformers import AutoTokenizer, AutoModel
from sklearn.metrics import accuracy_score
```

Listing 1: Necessary Packages to Import

After loading the data, the "label" column was encoded using one-hot encoding for compatibility with the models. The dataset was split into training and testing sets using an 80/20 split. A validation set was not created for this project due to time constraints.

Preprocessing involved tokenizing the "Title" column using the HuggingFace tokenizer. Key parameters included: - **return_tensors='pt'**: Ensures compatibility with PyTorch. - **padding=True**: Enables batched processing by padding sequences shorter than 512 tokens. - **truncation=True**: Truncates sequences longer than 512 tokens to ensure uniform input length.

These preprocessing steps ensured that the data was correctly formatted for BERT's input requirements.

Following this, an optimizer called AdamW, which is popular from pytorch, was assigned with an learning rate of 5e $^{-}$5. The learning rate is set to how much each iteration should update the weights for each parameter.

Finally, a for loop was setup to pass over the data 3 times in which AdamW optimized through the data. It is important to note that in this step, it was set to zerograd() which means that backpropigation gradients did not acrue for each pass. As this part is only extracting the features for the model gradients do not need to be calculated, from there outputs (features) are extracted from the inputs we set and then cls embeddings from the models hidden states are set to a numpy array.

```

tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased-
    finetuned-sst-2-english")
model = AutoModel.from_pretrained("distilbert-base-uncased-finetuned-
    sst-2-english")
inputs = tokenizer(news_df['title'].to_list(), return_tensors='pt',
    padding=True, truncation=True, max_length=512)
optimizer = AdamW(model.parameters(), lr=5e-5)
model.train()
for epoch in range(10):
    optimizer.zero_grad()
    with torch.no_grad():
        outputs = model(**inputs)
        cls_embeddings = outputs.last_hidden_state[:, 0, :].numpy()
```

Listing 2: Implementing and Fine-Tuning BERT

2.3 Model Architecture

When deciding on the parameters for the model building, it was important to consider computational time. For epochs, 1, 3, and 5 were tested and timed. Respectively each took, 3.5 minutes, 9.5 minutes, and 16 minutes. Additionally, when selecting learning rate, 3 were tested for changes in model accuracy. The learning rates for the AdamW optimizer were $5e^{-}9$, $5e^{-}5$, $5e^{-}1$. with respective accuracies of .765, .770, and .760. Indicating $5e^{-}5$ has the highest accuracy by a slight amount.

3 Experimentation Results

3.1 Main Results

After implementing BERT for tokenization, two classification models were selected to compare: Logistic Regression and KNN. The test error across the different epochs for logistic regression were generally higher than that for KNN as can be compared in figure 2 and figure 3.

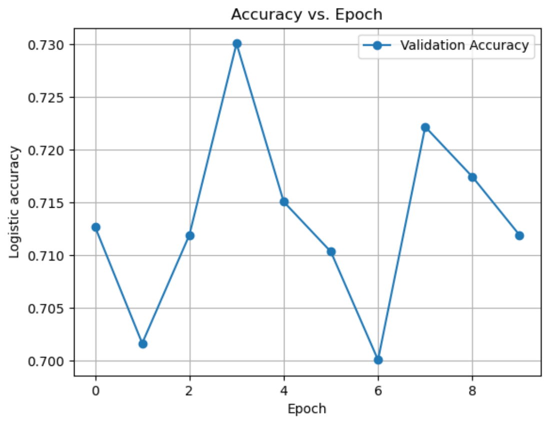


Figure 2: Test accuracies across epochs for logistic regression

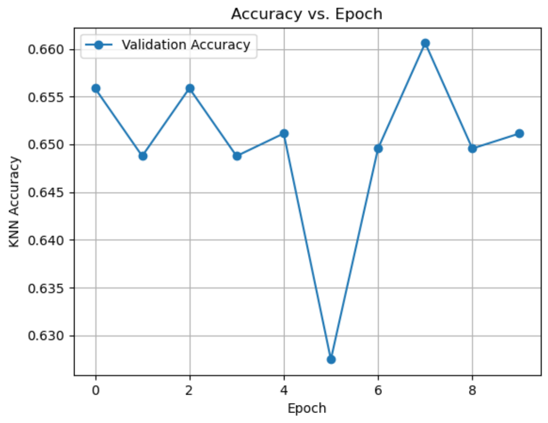


Figure 3: Test accuracies across epochs for KNN

For logistic regression, the most optimal epoch with the highest accuracy was 3, while the one for knn, it was 7.

For Logistic Regression, cross validation error was $_{acurracywas.77.}$

For KNN, cross validation error was $_{acurracywas_.}$

3.2 Supplementary Results

For final submission it

4 Discussion

This project used a distilled BERT uncased model combined with logistic regression and KNN to classify news titles as real or fake based on sentiment. Logistic regression outperformed KNN, likely due to its ability to handle the high-dimensional embeddings produced by BERT effectively, whereas KNN struggled with the cons of dimensionality. The optimal configuration of 3 training epochs and a learning rate of 5e-5 ensured the model reached peak accuracy possible. Distilled BERT, as a compact alternative to full-scale BERT, balanced computational efficiency and performance.

Despite promising results, the model's generalizability might be limited by the dataset size and fixed training epochs. Further exploration with larger datasets and additional training time could enhance robustness. KNN performance might benefit from dimensionality reduction techniques like PCA. Future work could also explore further fine-tuning of the distilled BERT model or the use of ensemble methods to boost accuracy. Expanding the model's application to full article text and integrating multimodal features such as metadata or images could further improve its utility for fake news detection.

5 Conclusion

This project successfully applied a distilled BERT uncased model combined with logistic regression and KNN classifiers to analyze sentiment in news titles for detecting fake news. The AdamW optimizer, with a carefully chosen learning rate of 5e-5, enabled efficient weight updates during training over three epochs, leveraging zero gradient accumulation for computational efficiency during feature extraction.

Future research could expand on this work by incorporating additional data, using more advanced fine-tuning techniques, and exploring ensemble approaches for improved accuracy and generalization. This project highlights the significance of combining advanced natural language processing models with practical implementation strategies for addressing contemporary challenges in information reliability.

At the conclusion of this project, team members gained an in-depth understanding of the real-life possibilities of machine learning models on different fields that affect the public directly. Subsequently, members have insights into self-attention mechanisms and what they mean.

6 References

References