CIS 4190/5190: Applied Machine Learning Fall 2024

**News Source Classification**

*Team Members: Feng Jiang, Jiayi Chen, Zihan Wang* *Project: News Source*

# 1 Summary

In this project, we developed a text classification model to classify news titles into their corresponding news sources. Before we went through this project, we raised an exploratory question: Does transformer-based models outperform the traditional machine learning models on text classification tasks?

We started with converting text data into TF-IDF features. This approach was simple and gave us an initial sense of the data. We applied it to models like Logistic Regression, Naive Bayes, and Linear SVM, which provided a solid baseline. However, we noticed that these models could not capture the deeper meaning of the words, so the performance was limited.

To address this, we used GloVe embeddings, which allowed us to represent words in a more meaningful way using pre-trained word vectors. By applying GloVe to models like Naive Bayes, SVM, and Logistic Regression, we saw a improvement of 10%. Among these, SVM achieved the best accuracy of 72% because it effectively optimized the decision boundaries and handled complex feature interactions.

To further improve, we switched to transformer-based models because they are powerful for understanding the context in text. We started with DistilBERT, which is a smaller and faster version of BERT, and then moved to RoBERTa. RoBERTa performed the best because of its better pre-training and dynamic masking strategies. After fine-tuning the hyperparameters like batch size, learning rate, and epochs using cross-validation, we achieved a final test accuracy of 0.8615 with RoBERTa. Compared to traditional machine learning models, we found that transformer-based models reached significantly higher accuracies, which answered our first exploratory question.

# 2 Core Components

## 2.1 Data Collection

For the data collection phase of our project, we were provided with a dataset consisting of URLs from two news sources: Fox News and NBC News. The goal was to extract the news titles from these URLs and label them based on their respective sources. This data would serve as the foundation for our news classification task. To automate the process, we developed a web scraping pipeline that extracted titles from the HTML content of each URL.

We used the requests library to fetch the web pages and BeautifulSoup to parse the HTML structure. Since the formatting of news headlines differed between Fox News and NBC News, we identified specific HTML tags and classes for each outlet. During the scraping process, we encountered several challenges. Frequent requests to the websites led to temporary blocks, which prevented us from accessing some URLs. To resolve this problem, we implemented random delays between requests using the time.sleep() function, introducing pauses of 5 to 10 seconds after every 50 URLs. Additionally, we rotated through a list of User-Agent strings to mimic requests from different browsers and reduce the likelihood of being flagged as automated traffic. To further optimize the process, we employed parallelization using Python's ThreadPoolExecutor, which allowed us to scrape multiple URLs simultaneously and significantly reduced the total runtime. After successfully processing the URLs, we labeled each entry based on its source: URLs containing ”foxnews.com” were labeled as 1 (Fox News), and those containing ”nbcnews.com” were labeled as 0 (NBC). This labeling was essential for preparing the dataset for binary classification tasks.

Once the scraping was completed, we curated and cleaned the data to ensure it was ready for further analysis. In total, we successfully processed 3805 URLs, resulting in a final dataset with four columns: url, title, outlet, and labels. After dropping duplicated titles, this dataset contained 3789 rows, with labels balanced between the two news sources.

For model development and evaluation, we split the dataset into training and testing sets. We used 80% of the data for training and 20% for testing to ensure a reliable evaluation of our classification models. The split was stratified to maintain an even distribution of Fox News and NBC labels across both sets. The final dataset provided a clean and labeled collection of news titles, which became the input for the text classification models explored in our project.

## 2.2 Model Design and Evaluation

### 2.2.1 Baseline model

We started our project by loading the dataset, which included news titles and outlets. To prepare for modeling, we renamed the relevant columns and transformed the outlet names into binary labels: '1' for 'FoxNews' and '0' for 'NBC'. We split this data into training and testing segments. For text processing, we applied TF-IDF vectorization, limiting ourselves to the top 100 features and excluding common English stopwords. Our baseline consisted of traditional models—Logistic Regression, Naive Bayes, and Linear SVM. We trained and tested these models to establish a performance benchmark that would help us assess more complex models later in the project.

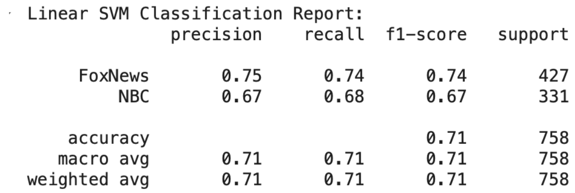
For the logistic regression model, we utilized Scikit-Learn's LogisticRegression with a maximum of 100 iterations, achieving an accuracy of 0.6741 by fitting and predicting on TF-IDF vectors. Similarly, the Naive Bayes model, implemented via MultinomialNB, matched the logistic regression's accuracy, performing slightly better in recall for class 1. The Linear SVM, using LinearSVC with up to 1000 iterations, provided a marginal accuracy improvement at 0.6781. These baseline models establish initial performance benchmarks and inform further refinements in our model development strategy.

### 2.2.2 Data Preprocessing

The initial step involved a thorough examination of the dataset to ensure there were no missing values, and we found 16 duplicate headlines which were removed to improve the dataset's quality. For preprocessing, we focused on standardizing the text by converting it to lowercase and expanding contractions, which made the analysis more consistent. We also removed monetary symbols, URLs, and special characters to reduce potential noise in the data. These steps were crucial in refining our data set, as indicated by the reduced number of headlines with special characters.

### 2.2.3 GloVe embeddings

We chose GloVe embeddings for our project due to their unique ability to capture both global statistics and local context of words within a corpus. GloVe effectively combines the advantages of two model families: the global matrix factorization and local context window methods. We chose to use the GloVe.6B.100d embeddings to balance between computational efficiency and the complexity of the model. While exploring GloVe.6B.200d or higher models, we found that they yielded the same accuracy. The use of embedding markedly improved the metrics of the above-mentioned three models, boosting precision, recall, and F1 scores by an average of 0.1. Given that TF-IDF or word embeddings creates subtle relationships and patterns within the high-dimensional feature space, we figured out SVM, by optimizing the decision boundary, better captures these subtle interactions, resulting in higher precision, recall, and F1-scores, as reflected in the classification report.



```
Linear SVM Classification Report:
              precision    recall  f1-score   support

     FoxNews       0.75      0.74      0.74       427
         NBC       0.67      0.68      0.67       331

    accuracy                           0.71       758
   macro avg       0.71      0.71      0.71       758
weighted avg       0.71      0.71      0.71       758
```

Figure 1: Classification Report for SVM with GloVe Embeddings

#### 2.2.4 Transformer

Realizing the potential for even greater accuracy, we employed the transformer architecture, specifically the DistilBert model, as our primary choice. DistilBert, a simplified version of the more complex BERT architecture, is well-suited for processing and classifying text while being more efficient. DistilBert retains much of BERT's effectiveness but is lighter and faster, making it ideal for our project where we aimed to classify news titles into specific categories. This was crucial for our task, which demanded precision in understanding subtle linguistic cues in the text.

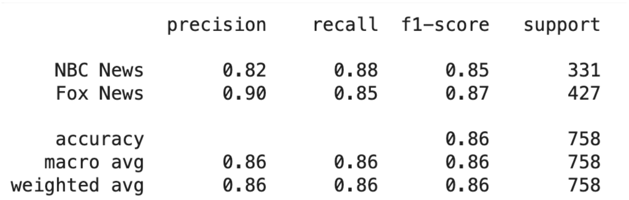
For setting up the model, we began with the 'distilbert-base-uncased' configuration, incorporating a tokenizer that processes the text into a structured format that the model can interpret. This involves transforming the text into input IDs and attention masks, which help the model focus on significant parts of the text sequences. We trained our classifier through several iterations, continuously optimizing it to reduce errors and enhance its predictive accuracy. By using custom Dataset and DataLoader classes, we managed the data flow during training efficiently, ensuring the model received properly batched and shuffled data. The culmination of these efforts was reflected in our final test accuracy of 0.8463.

Consider that RoBERTa employs dynamic masking, where the masked tokens vary across epochs, allowing the model to learn richer contextual representations, we employs RoBERTa models, each configured to manage two labels. Data preparation involved using custom Dataset and DataLoader classes. The model was previously trained using the AdamW optimizer with a learning rate of 3e-5 over three epochs. During evaluation, the model achieved a Test Accuracy of 0.8496, which performs slightly better than BERT model.

After observing the potential improvement in the RoBERTa model, we fine-tuned our RoBERTa model by testing multiple combinations of hyperparameters, including batch size, epochs, and learning rate. After evaluating the model across 4 folds, the optimal configuration was identified as a batch size of 32, learning rate of 3e-5, and 4 epochs, achieving the best mean CV score of 0.8541. Applying this configuration to the final training process, the RoBERTa model achieved a final test accuracy of 0.8615, which is also the final model we pushed to huggingface.

#### 2.2.5 Confusion Matrix

We constructed a confusion matrix to present the evaluation. The RoBERTa model achieves an overall accuracy of 86%, showing strong performance with some room for improvement. For NBC News, it correctly identifies most examples, reflected in a high recall of 88%, though its precision is slightly lower at 82%, indicating some misclassifications as Fox News. On the other hand, for Fox News, the model demonstrates a high precision of 90%, meaning its predictions are mostly correct, but the recall drops slightly to 85%, as a few instances are misclassified as NBC News. In total, 40 NBC News and 65 Fox News examples were misclassified, suggesting the model is generally balanced.



```
              precision    recall  f1-score   support

    NBC News       0.82      0.88      0.85       331
    Fox News       0.90      0.85      0.87       427

    accuracy                           0.86       758
   macro avg       0.86      0.86      0.86       758
weighted avg       0.86      0.86      0.86       758
```

Figure 2: Classification Report of Final RoBERTa Model

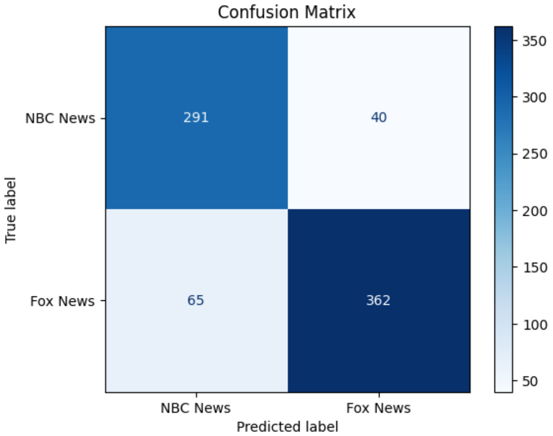


Figure 3: Confusion Matrix of Final Model

# 3 Exploratory Questions

## 3.1 Question and Motivation

*"How Do Different Parts of Speech and Token Types Influence RoBERTa's News Classification Decisions?"* This question examines how linguistic components in news headlines affect the model's decisions. Understanding this helps identify which parts of speech are most important, uncovers potential model biases, and informs both headline writing and model improvements.

## 3.2 Prior Work and Expectations

Our investigation builds on Clark et al.'s (2019) paper "What Does BERT Look At? An Analysis of BERT's Attention." Their work is particularly relevant as RoBERTa builds upon BERT's architecture. The study revealed that BERT's attention heads develop specialized functions, with different heads focusing on specific parts of speech (POS), dependency relationships between words, previous/next token relationships, and punctuation marks. Based on these findings, we expected high attention to nouns and proper nouns, moderate attention to verbs, low attention to articles and prepositions, and a significant role of punctuation in attention patterns.

## 3.3 Methods for investigation

Our investigation employed several key computational techniques to analyze how different parts of speech influence news classification. We used spaCy, a natural language processing library, to identify word types in headlines. Words were categorized into distinct groups: nouns, proper nouns, verbs, adjectives, numbers, and other elements like punctuation. To measure word importance, we analyzed attention weights from our RoBERTa model, which indicate how much the model focuses on each word during classification. Finally, we created visualizations using matplotlib and seaborn to display word frequency distributions, importance rankings, and token influence patterns.

### 3.4 Results and Updated Beliefs

Our analysis revealed patterns in RoBERTa's processing of news headlines, with proper nouns emerging as the dominant force in classification decisions. The importance scores were particularly revealing: names and places like "Gaza", "Trump", "Biden", and "Harris" showed significantly higher influence than other word types. This hierarchy of importance extended through different grammatical categories, with proper nouns scoring 2.3, regular nouns 2.1, and verbs 1.8, establishing a clear pattern of how the model prioritizes different types of words in making classification decisions.

Perhaps most intriguingly, our analysis revealed several unexpected patterns that challenged our initial assumptions. Punctuation showed a surprisingly high importance score of 2.6, suggesting that structural elements play as crucial a role as content words in classification. Function words like "for", "says", and "after" ranked among the top 10 most influential tokens, indicating that these connecting words carry more semantic weight than traditionally assumed. The distribution analysis showed a heavy skew toward nouns and proper nouns, contrasting with our initial expectation of a more balanced distribution across word types. These findings suggest that news classification models rely on a more complex interplay of linguistic features than previously understood, with both content and structural elements working together to determine article categorization.

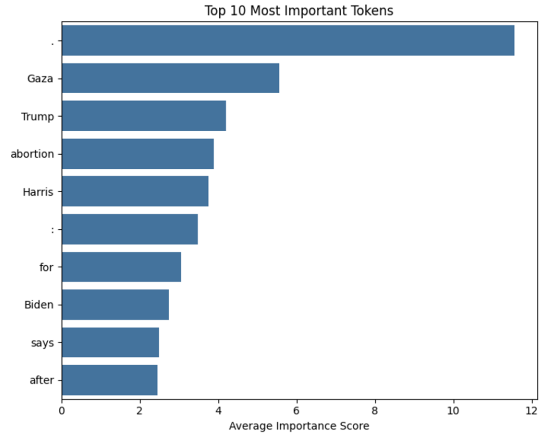


Figure 4: Top 10 Most Important Tokens

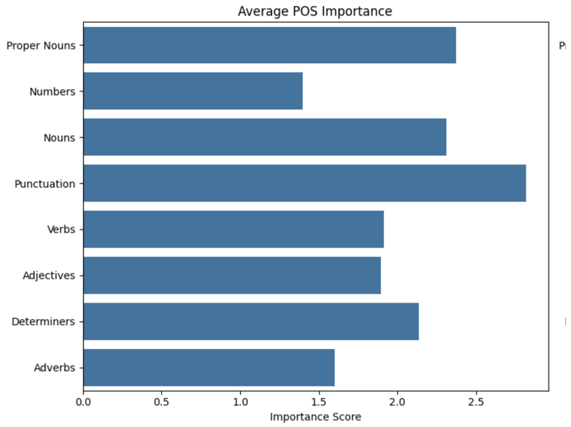


Figure 5: Average POS Importance

### 3.5 Limitations and Future Work

The analysis was limited to English-language headlines and binary classification tasks. Expanding to include multiple languages, news categories, and larger datasets would provide deeper insights. Exploring interactions between parts of speech and temporal patterns could also enhance understanding of linguistic influences on classification decisions.

## 4 Team Contributions

Feng Jiang: Data collecting, transformer-based model (BERT & RoBERTa) building, project report writing.
Jiayi Chen: Data cleaning and preprocessing, cross validation and hyperparameter tuning, exploratory question: influence of speech and token.
Zihan Wang: Baseline models building, embedding building, DistilBert hyperparameter tuning, project report writing: summarize and evaluate the models.

## Acknowledgements and Citation

We acknowledge the work by Clark et al. (2019), whose paper *"What Does BERT Look At? An Analysis of BERT's Attention"* inspired our investigation into RoBERTa's attention mechanisms for news classification. Their findings provided foundational insights into how attention heads specialize in focusing on parts of speech. The full article is available at: https://arxiv.org/pdf/1906.04341.

Additionally, our final fine-tuned RoBERTa model has been published on Hugging Face. You can access it here: https://huggingface.co/CIS5190GoGo.