Hyperpartisan Article Detection - Model and Dataset Analysis

Mark Glinberg
mglinberg@gatech

Stephan Tserovski
stserovski3@gatech.edu

Justin Huang
thuang348@gatech.edu

1 Introduction

With the U.S. Presidential Election looming, we wanted to work on a project that could shed some more light on voter opinions in order to predict the election outcome. As such, our original project as stated in the proposal, was to design and build a dataset of Tweets related to the 2024 U.S. President Election. Our goal was to define annotation guidelines and use this dataset to predict election results based on overall sentiment towards a certain party, and maybe even have our dataset be used as a good training source for future models. However, as we delved deeper into this endeavor, we encountered monetary barriers with using the X API, as well as feasibility barriers with defining proper annotation guidelines for sensitive topics such as political discourse. As such, we decided to pivot our project to instead focus on hyperpartisanship in the media.

In the contemporary political setting, especially during election seasons, the internet becomes flooded with hyperpartisan content, often characterized by articles that convey extreme biases to influence public opinion. Articles like that can pigeonhole people into certain beliefs without adequately providing people nuanced viewpoints and opposing opinions. To help combat this, we pivoted our project to focus on analyzing an existing approach to determining hyperpartisanship in media. This work is crucial in understanding the categorization of information and its impacts on the internet.

The model studied in this project makes predictions based on a comprehensive analysis, incorporating both article titles and text. By exploring the strengths and weaknesses of the model, there's an opportunity to refine the technical aspects of natural language processing and its accuracy in identifying hyperpartisan content. The motivation behind enhancing this model is to create a robust tool that can recognize political rhetoric and provide a clearer picture of the media landscape during critical times such as elections.

2 Related Work

The model we analyzed was based on DistilBERT, a transformer-based model with attention mechanisms which we used to analyze the performance of our machine learning model based on the work done in Ballout et al. (2023). This particular research paper provides a comprehensive analysis of transformers, specifically BERT's attention mechanisms, and their superiority over traditional models like RNNs. By focusing on non-language tasks such as arithmetic operations, the study examines how transformers assign different importance across an input sequence using attention weights. The research also introduces several visualization methods to understand the attention mechanisms within BERT when applied to non-language tasks. While their visualization methods weren't the exact ones we utilized, they align with the fundamental concept of utilizing transformers' unique properties to explain a model's predictions. These methods help illustrate the internal decision-making processes of the model, and reveal patterns in how the model attends to different parts of the input. Through these techniques, the paper demonstrates how the model can generalize to longer sequences and how specific layers of BERT contribute to task performance.

In our project, we implemented LIME visualizations to identify and analyze the presence and impact of hyperpartisan language within text datasets. In their paper, Mardaoui and Garreau provide a theoretical examination of the LIME (Local Interpretable Model-agnostic Explanations) method as applied to text data. They discuss the effectiveness of LIME in providing meaningful explanations for machine learning models, particularly when handling text data. The authors focus on ensuring that LIME can reliably interpret these models and their analysis addresses how LIME samples and

weighs features from text data. The visualization provided by LIME is essential in interpreting machine learning models, allowing for insights into which features and words are driving the predictions (Mardaoui and Garreau, 2021).

3 Methodology

For our project, we utilized the Hugging Face API to choose a dataset containing over 750,000 articles, each categorized as either hyperpartisan or neutral. We consider the size and quality of this dataset to be substantial enough to conduct reliable analysis on. After acquiring the dataset, we imported a DistilBERT based natural language processing model which reported a 99% accuracy on the training data. Our goal was to test this accuracy by applying the model to the validation data and examining the results to gain insights on the limitations of the model's predictions. When we tested the model using a concatenation of the validation article titles and texts, we discovered a significant drop in model accuracy. We found that the model clearly struggled with limited information and performed better when given longer titles and texts. Furthermore, our analyses on the model's attention weights and other features of the dataset were able to reveal various biases which we believe to have impacted performance.

To make these trends more comprehensible, we utilized matplotlib to graphically represent the relationship between text length and accuracy, as well as the model's predictions in relation to the political labels of the articles. We also used a confusion matrix to visually represent the model's tendency to mark articles as hyperpartisan. Finally, we employed techniques such as LIME to identify specific biases in the model such as the model's tendency to overemphasize politically charged words. This analysis was important in recognizing the areas where the model might misinterpret the data, allowing us to better understand the limitations of the model and improve upon it. After conducting our evaluation, we compared our findings to other research done in this specific problem domain.

4 Dataset

The dataset we found on Hugging Face is structured for text classification to detect hyperpartisan news content. It includes over 750,000 articles with information on each one including the title, text, publisher's political bias, publishing date, article's URL, and whether or not the article is hyperpartisan. The title entry includes the title of the article while the text entry contains the content of the news article. The bias entry classifies the article's publisher as either right, right-center, least, left-center, or left. The publishing date and URL are self-explanatory and the hyperpartisan field is a boolean indicating whether or not the article is considered extremely polarized (Kiesel et al., 2019). There is an equal distribution of neutral and hyperpartisan articles in the dataset as seen in Figure 1.

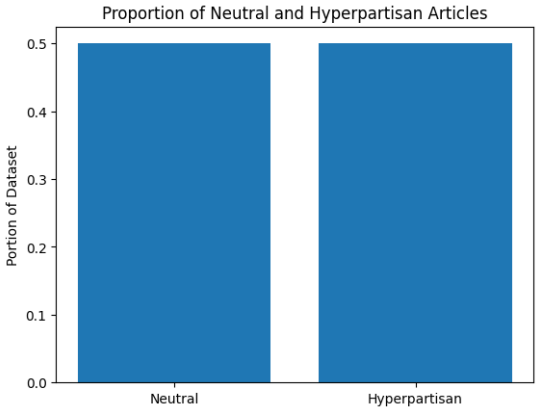

Figure 1: Even distribution of Neutral and Hyperpartisan articles in the dataset.

5 Experimental Results and Findings

As stated in our methodology, we ran the model on the validation set containing 150,000 entries. The model got an accuracy of 0.58, with a precision of 0.55, recall of 0.83, F1 Score of 0.66, and a false positive rate of 0.67, all calculated from the confusion matrix in Figure 2.

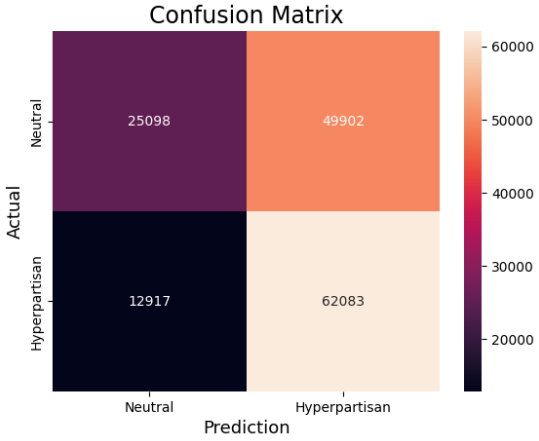

Figure 2: Model output Confusion Matrix values out of 150,000 entries in the dataset.

These results were significantly worse than expected, based on the self-reported accuracy of the model being 0.99. When running the model on the train dataset, we did in fact see an accuracy of 0.99, meaning that most likely the model was severely overfitted on the training data. However, despite these poor results, there's still a lot we can learn from how this model behaved. Based on its high false positive rate and high recall, we can see that the model has a very high tendency to output false positives and labels most articles as hyperpartisan, regardless if they actually are.

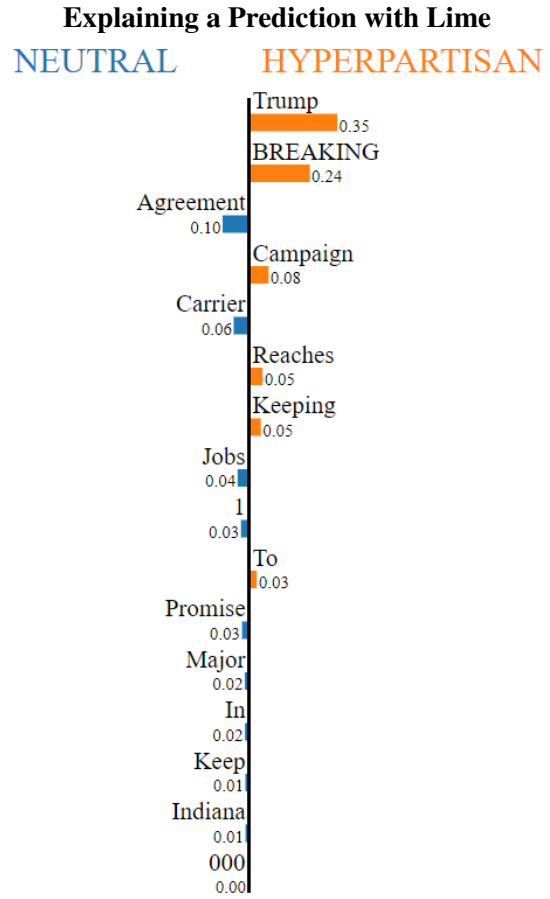

Figure 3: Using LIME to explain the model's prediction of the headline "BREAKING: Trump Reaches Agreement To Keep 1,000 Carrier Jobs In Indiana, Keeping Major Campaign Promise." Hyperpartisan words are orange while neutral words are blue and the words are all ranked by their values.

Diving deeper into our analysis, we can see in Figure 3 that the model viewed the words Trump and BREAKING as very hyperpartisan. This makes sense considering that Trump is a very politicized figure and breaking news can often sacrifice nuance in exchange for releasing information as fast as possible. However, this labeling means that articles which neutrally discuss Trump could be falsely marked as hyperpartisan. Breaking new articles likewise, are also not always hyperpartisan, and making trivial assumptions like these most likely contributed to the model's high false positive rate.

The results from transformers-interpret (Figure 4) generally align with the explanation from LIME. The key similarities include a clear emphasis on the words Trump, breaking, agreement, and campaign. However, it seems that transformers-interpret finds a greater influence from some words such as major and jobs. Cross referencing the two methods provides us with a more complete explanation of the model's predictions. Since there is a clear common emphasis on Trump, breaking, and agreement, we can determine that these words are very important to the model's prediction of this title.

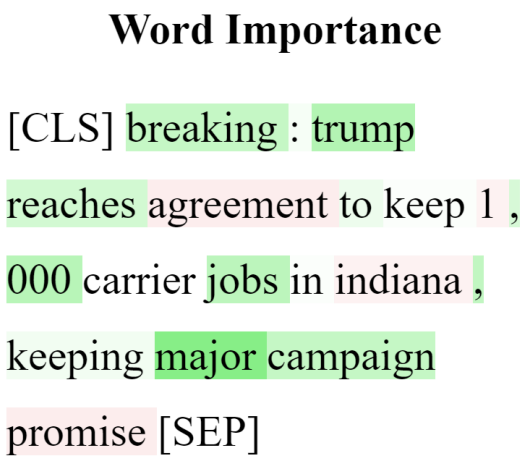

Figure 4: Using transformers-interpret to explain the model's prediction of the headline. The words which contribute to the hyperpartisan prediction are highlighted in green. The words which influence toward a neutral prediction are highlighted in red.

Beyond statistics and attention weights, we also looked at how the length of the article titles and bodies impacted the model's performance. As you can see in Figure 5, the top two charts show the distribution of the title and text lengths of the articles the model labeled correctly, and the bottom two charts represent the same distributions but for incorrect labels. Based on these charts, the model shows a slight improvement in classifying articles with longer titles and bodies, which is most likely due to having more information to work with. This relationship between text length and accuracy highlights the model's dependency on sufficient data to parse and understand the language used in hyperpartisan content. This dependency points to the

need for models that can operate efficiently and effectively across a variety of text lengths and complexities.

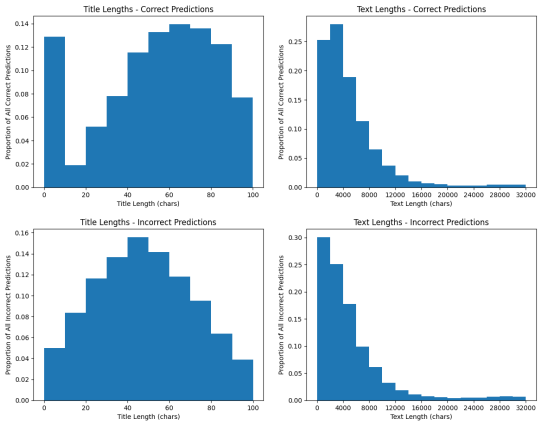

Figure 5: Title and Text Lengths relative to the model's correct and incorrect predictions.

Using the bias labels to analyze the model's labels (Figure 6, we can reaffirm the model's tendency to label most articles as hyperpartisan. The "left" and "right" articles (which are also labeled as hyperpartisan) have a very high correct prediction rate, while the more center-leaning articles are much more likely to be mislabeled. On top of that, the model exhibited a slight bias towards labeling left-leaning articles as hyperpartisan more frequently than right-leaning ones. This notable difference indicates a reevaluation of the training data is necessary to ensure a balanced representation of political perspectives, which is important for unbiased media analysis.

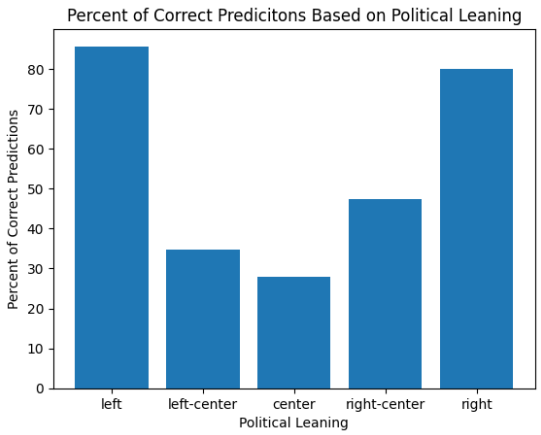

Figure 6: Analysis of the influence of political leaning on the model's success. Hyperpartisan leanings are much more likely to be classified correctly, and left-leaning articles are more likely to be classified as hyperpartisan.

6 SOTA Comparison

Palić et al. (2019) suggest in their paper that incorporating the publishing date of articles can significantly enhance the accuracy of models designed to detect hyperpartisan content. This approach is based on the observation that articles published during or near election cycles may be more likely to be hyperpartisan. The context of an article's publication can serve as a significant indicator of its content's nature, particularly during election cycles when partisan content is more common. With that said, each article in our dataset is tagged with its publishing date, which is valuable information as indicated. These findings imply that the model we analyzed may have had improved accuracy if it took this additional factor into consideration. However, it also might detract from our overall goal of sorting through articles relevant to the election, regardless of publication date, and helping voters find neutral sources of information. Having a model that rules out all articles published around election cycles wouldn't provide the nuance needed to solve this problem.

Additionally, Palić et al. highlight potential benefits from incorporating a "trigger words" dictionary into the model. This approach could theoretically enhance the model's accuracy by allowing it to more easily identify words with strong political or emotional connotations. While the analyzed model's lack of such a dictionary may have contributed to its worse performance, our findings suggest that this may not be a foolproof solution. Specifically, our observations indicate that the model may disproportionately focus on politically charged words, such as "Trump," skewing its predictions toward an overestimation of partisanship in articles where such terms appear. One example of how this method could backfire is if an article uses such a "trigger word" in a historical or analytical context, but the model falsely identifies it as hyperpartisan. Therefore, while the approach will likely provide accurate predictions in a majority of contexts, our findings indicate that safeguards would need to be put in place to prevent inaccuracies.

A unique aspect of our results is our focus on the length of the text and title. Existing research predominantly focuses on the effects of data scarcity on model performance. On the other hand, our investigation presents a novel angle by demonstrating the benefits of lengthier text data in improving

accuracy. These insights are valuable as they encourage further research into making models more robust against the challenges posed by limited data.

7 Conclusion

Our exploration into hyperpartisan content detection through natural language processing has provided valuable insights and highlighted significant challenges in hyperpartisan detection. Despite initial high expectations, where models proclaimed near-perfect accuracy, our evaluations revealed a significant decline in performance when presented with limited and novel textual information. This highlights the role of content length and detail in achieving reliable classification outcomes.

Looking forward, there are a few different avenues that can be followed to push forward possible solutions for this problem. To start, this existing approach may be improved by increasing the dropout rate to reduce overfitting, and in turn enhancing the model's ability to generalize across varied datasets. Going beyond adding a dropout layer, experimenting with different model architectures will be important in identifying the most effective framework for hyperpartisan detection. Finally, examining completely different approaches could prove to increase the overall performance of such models. As we continue to refine these tools, our goal is to contribute to the development of a more balanced and accurate system for media analysis that is capable of operating effectively regardless of text length or the political leanings of the content.

Limitations

First, this is a fairly novel problem, and as such, there aren't many datasets for it or available models to solve it. As such, it's difficult to have a robust and all-encompassing analysis with such few resources to work with. As this problem is explored further, more data can be gathered and new models can be developed and refined to better solve this problem.

Beyond the novelty of the problem, while we used tools to address the black-box nature of NLP models, we still don't have a completely clear picture of how the model made its decisions. The tools we used gave us insight into how much the model viewed certain words as hyperpartisan or not, but that could change depending on the context of the words themselves, which is harder to analyze. The black box problem is an inherent problem to all neural networks, and ideally more tools need to be developed in order to delve deeper into all models' biases in order to strengthening our understanding of how to improve their performance.

Finally, a problem like this requires many, many hours of research and development, and we did what was feasible in the course of one semester. To really tackle and solve this problem, researchers need spend significantly more time investigating existing solutions and then using those results to incrementally improve upon those solutions, until a working model is developed.

Contribution Table

Contribution	Contributors
Picking a Dataset	Mark, Stephan, Justin
Picking a Model	Mark, Stephan, Justin
Testing the Model	Justin
Creating Visualization Graphs	Mark
LIME and transformers-interpret	Stephan
Presentation	Mark, Stephan, Justin
Final Report	Mark, Stephan, Justin

References

Mohamad Ballout, Ulf Krumnack, Gunther Heidemann, and Kai-Uwe Kühnberger. 2023. Opening the black box: Analyzing attention weights and hidden states in pre-trained language models for non-language tasks.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Dina Mardaoui and Damien Garreau. 2021. An analysis of lime for text data. In *International conference on artificial intelligence and statistics*, pages 3493–3501. PMLR.

Niko Palić, Juraj Vladika, Dominik Čubelić, Ivan Lovrenčić, Maja Buljan, and Jan Šnajder. 2019. TakeLab at SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 995–998, Minneapolis, Minnesota, USA. Association for Computational Linguistics.