# Media Bias Article Classification using Supervised and Unsupervised Learning

Gabrielle Jones*, Jessica Ma†, Shivam Patel‡, Wesley Tsai§, Tao Zhou¶
Emails: {*gabs, †jqma, ‡shivamgp, §wesleyjt, ¶taozhou}@umich.edu

EECS 545 - Machine Learning
University of Michigan
Ann Arbor, MI USA

***Abstract*—News and general media have never been more accessible than before, and likewise it has never been easier for highly biased media to dominate the market. Within the United States two regimes ("left" and "right") dominate the political bias spectrum, leaving consumers who want objective ("neutral") news stranded. Recent improvements in artificial intelligence models have been garnering attention as a potential "objective" rating system, one not susceptible to human's unavoidable bias. Such a system would increase consumer trust in media and work as an effective filter for identifying unbiased sources. This work's supervised learning methods demonstrate levels of classification accuracy up to 80% using a GPT-2 architecture, pivoting from its common use of text generation. A probabilistic supervised Naive Bayes method showed that predicting bias using text tokenization are category dependent; a no correlation regime, a highly context dependent regime, and a final "biased phrase" dominated regime. Unsupervised learning methods, free from external human bias also showed improvements within our findings, as compared to a random guess baseline. A novel unsupervised classifier based on OpenAI's Chat API achieved a 46.8% accuracy, with a high internal bias for classifying articles as "neutral". These findings suggest that unsupervised methods may cater towards neutrality and overlook satire, shedding insight into their mechanisms for specialized tasks, all reflected against our society's expectation for AI to capture the full human experience. Political bias is a spectrum, dependent on day-to-day events, and this work presents new insights into how supervised and unsupervised methods attempt to parse such a culturally nuanced task.**

## I. Introduction

With today's media oversaturation, the question of embedded bias is increasingly important to everyday readers and prospective writers. The global pandemic and recent political climate have brought media bias and "fake news" conversations to the forefront of everyday consumers' minds. Polarization and bias in the media often influences public opinion, swaying elections, and ultimately affecting public policy [1]. It is difficult for writers and readers to know which outlet to trust; writers have difficulty tailoring writing towards specific audiences, modifying bias to increase the chance of acceptance, while readers often cannot or do not have the time to accurately discern the levels of bias present. Many suggestions are given to readers such as avoiding the use of the word "I", the use of personal attacks, and the presentation of facts without supporting evidence [2]. There are other filtering methods taught known by the abbreviation *VIA*: is the information *verifiable*, *independent*, and *accountable*? These two concepts can't solve the problem that is people's lack of desire to to find objective news sources, known as audience bias. Furthermore, consumers tend to believe that the majority of news media is biased against their own political alignment, and those who frequently discuss media bias often have skewed views of what media is truly unbiased [3]. We'll be focusing on tackling the front end of bias which can take out the activation energy for people as they could potentially filter articles in an automated way based on bias level. This could allow an "objective" rating system of news articles that audiences could check before they consume a piece of media, leading to increased trust in news outlets and a more objective view of the political climate.

## II. Related Work

One reliable method of determining the media bias of an article is to examine the publisher or source that it was derived from. Many websites such as "AllSides" and "Media Bias/Fact Check" utilize expert bipartisan reviewers to look for slant, spin, sensationalism, and story choice, in different news outlets. Furthermore, they poll civilians on the media bias of these same outlets in addition to independent review. Unfortunately, these methods are extremely time-consuming and labor-intensive. In addition, they primarily label the bias of the media outlet, rather than the individual articles, which may falter from the political leaning of the news outlet as a whole, and the bias classification on these websites is limited to select media sources. Many consumers are increasingly getting news from social media, and these hand-picked baises cannot be extrapolated to such platforms because they host a wide variety of people and perspectives.

Most recently, researchers at MIT developed a Singular Value Decomposition (SVD) model to classify phrase bias, the left-right bias of certain phrases when talking about a given topic. For example, when talking about the "BLM" movement, right-wing media would use "riots" and left-wing media would describe use "protests" or similar words [4]. One flaw of this approach is that in order to classify the bias of an article, this SVD model needs to know the article topic. This means that if the model is not trained to understand the biases of a given topic, then it cannot classify the bias of an article.

In machine learning for natural language processing (NLP), transformers are one of the most recent and powerful developments. BERT, developed by Google, is a pre-trained model that can easily be fine-tuned for many different NLP tasks. It utilizes Masked Language Modeling (MLM), which involves masking a portion of a triaining text and trying to predict the masked words, to learn context from both directions of a token [5]. GPT-2, developed by OpenAI, is a transformer model that was designed to generate text word by word, predicting the next word in a sequence. This allows it to generate texts based on input prompts [6].

The approach we use in this paper compares supervised simple methods (Naive Bayes) and existing state-of-the-art methods (BERT and GPT-2) to the unsupervised GPT-3.5. Released late last year, the full potential of GPT-3.5 and the ChatGPT interface has yet to be fully explored. While fine-tuning of large language models such as BERT have been used to classify biases before [7], to our best knowledge, few if any papers have been published examining the ability of GPT models to classify media bias using either supervised or unsupervised methods.

## III. Method

In this section, we outline the methods used for the design, training, and evaluation of each of our machine learning models. The approaches we took for this project involved both supervised and unsupervised learning. For supervised learning, we trained a BERT and GPT-2 Transformer based text classifier as well as Naive Bayes approach. For unsupervised learning, we experimented with a ChatGPT-based approach.

The goal of each method provided is to evaluate the classification accuracy of these models when limited to low amounts of labeled training data. This is to avoid injecting human bias into the training data labels and therefore into the model training. Limiting training data is known to lead to overfitting issues in supervised models and we made sure to avoid overfitting as best as possible.

One important aspect to keep in mind with these methods is that we are not considering the effects of pre-training bias from the large text datasets used to pre-train the BERT, GPT-2, and ChatGPT Transformer models. There will always be some inherent bias when using a Large Language Model (LLM) for any text classification task but we are hoping these methods and evaluation help us find models and approaches to limit the further effects that humans have on the classification of text.

All code and datasets for these methods can be found on our GitHub repository at: Media-Bias-Article-Classification.

### A. Dataset

The dataset used for this project was the `MBIC` Media Bias Annotation Dataset from Kaggle [4]. This dataset included labels for approximately 1600 articles for political bias ("left", "center", "right"). Text was parsed using `Newspaper3k` [8]. From this dataset, 101 articles and their labels were randomly selected and withheld as our testing dataset for usage with all our final models. Raw truncated article entries and their labels from this dataset are included in Appendix A.

### B. Supervised Article Classification

The first approach we took involved supervised training by fine-tuning a BERT and GPT-2 Transformer based text classifier.

The same minimal pre-processing was done for each of the supervised classifiers. All duplicate articles were removed, articles were converted to only contain utf-8 characters, and all non-alphanumeric characters were removed. This pre-processing was minimally performed to maintain as much context as possible in the text between words. We experimented with the removal of stopwords as well (eg. "the", "is", and "and") and found a significant reduction in classification accuracy for both the BERT and GPT-2 classifiers. This was primarily explained by both transformer models creating text embeddings that relied on the relationships between keywords in phrases and sentences rather than the keywords themselves (expected in a Naive Bayes classifier for example). As a result, the stopwords were maintained to create better text embeddings.

*1) BERT Transformer Classifier:* For the BERT text classifier, we were looking to re-implement and possibly improve the accuracy achieved by Simoes and Castanos from Stanford University in 2020 when BERT text classification for large articles was initially gaining interest [7]. In their paper, these researchers were able to achieve approximately 68% test accuracy using a Huggingface "BERT Base Uncased" model [5].

This model was created using a Huggingface `BERTForSequenceClassification` model. This model consisted of a "BERT Base Uncased" model appended with two fully connected classification layers. These classification layers would be fine-tuned based on the training data provided for text classification into our desired classes

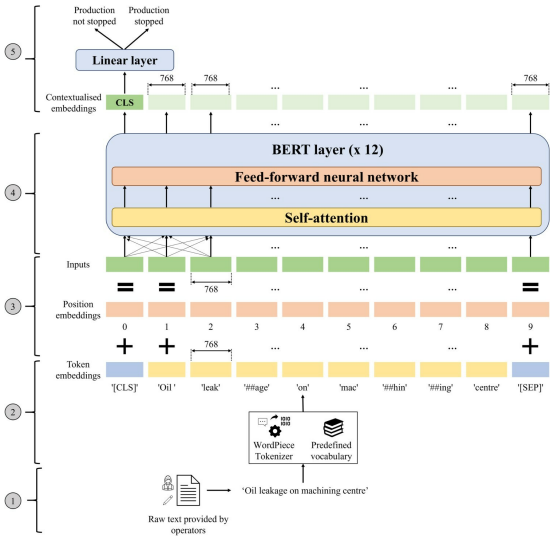


Fig. 1. BERT Text Classification Model Diagram [9]

("left", "center", "right") after a Softmax operation. This model diagram can be visualized in *Figure 1*.

In our fast and experimental re-implementation of this paper by fine-tuning a "BERT Base Uncased" model, we were able to achieve approximately 66% test accuracy similar to the results achieved in the paper training with a 90%/10% train/-validation split and 101 testing examples that we withheld for final testing. However, after re-implementation, we noticed our model was over-fitting very quickly when training with our relatively small dataset since the "BERT Base Uncased" has 110 million parameters. We were achieving close to 93% training accuracy with 65% validation and 68% using an Adam optimizer with learning rate decay after convergence. However, during training, we were reaching validation accuracy close to 88% before observing overfitting.

In order to improve text classification accuracy, our team decided to reduce the complexity of the model by employing a newer "DistilBERT Base Uncased" model [10]. We also added dropout at the last fully connected classification layer and weight decay during training to further reduce the effects of overfitting. A breakdown of the entire model architecture is provided in Appendix B.

This new "lighter" model was then trained with the same 90%/10% train/validation split and 101 testing examples we withheld for final testing. The hyperparameters for this model were then tuned using `SigOpt` for maximum validation accuracy. The hyperparameters we chose to tune were the weight decay and learning rate. We also tuned the dropout rate at the classification layer manually. This approach allowed us to achieve a significant improvement over the "BERT Base Uncased" model as discussed in our results.

Additionally, the goal of this project was to find ways to perform political bias detection without injecting too much human bias into training data. With this goal in mind, we re-trained our new "DistilBERT Base Uncased" model and tuned the hyperparameters and dropout rate for a variety of train/validation splits from 90%/10% down to 10%/90%. The hyperparameters and dropout needed to be re-tuned for each train/validation split to avoid overfitting to the smaller training datasets. We were unable to use the pre-trained model as is (with zero training data) for a fully unsupervised model due to the randomly initialized fully connected classification layers before fine-tuning. We performed this analysis with a variety of train/validation splits to evaluate the classification accuracy as related to the training dataset size to minimize the effects of human bias on article text classification for political bias by minimizing the required training data. These results were then compared with the GPT-2 Transformer based text classifier.

*2) GPT-2 Transformer Classifier:* For the second supervised model, we fine-tuned a GPT-2 Transformer based text classifier for poltical bias detection in articles. This method was pursued due to the recent surge in interest for GPT based transformer models and our inability to find any existing research on GPT based text classification for our application of political bias detection specifically on large articles (rather than sentences). Our research indicated that GPT based transformers are mostly used for text generation but can be used for text classification. Specifically, most GPT based text classification was done for short text entries (such as tweet and movie review sentiment analysis) and we wanted to take advantage of the ability for GPT-2 to handle up to 2048 token sequences at a time compared to BERT (and inherently DistilBERT) only handling 512 token sequences at a time. This presents an opportunity for GPT-2 to recognize long-term context relationships in text better than BERT. Due to our application of large article text classification, we believe GPT-2 could provide improvements over state of the art BERT-based text classifiers for these larger 1000+ token text inputs. This model diagram before the fully connected classification layers can be visualized in *Figure 2*.

The fine-tuning approach taken for the GPT-2 Transformer text classifier was very similar to the approach taken for fine-tuning the DistilBERT Transformer classifier. A Huggingface `GPT2ForSequenceClassification` model was initialized using a pre-trained GPT-2 Transformer. This model consisted of a Huggingface "GPT-2" model [6] appended with two fully connected classification layers. These classification layers would be fine-tuned based on the training data provided for text classification into our desired classes ("left", "center", "right") after a Softmax operation. Additionally, we added

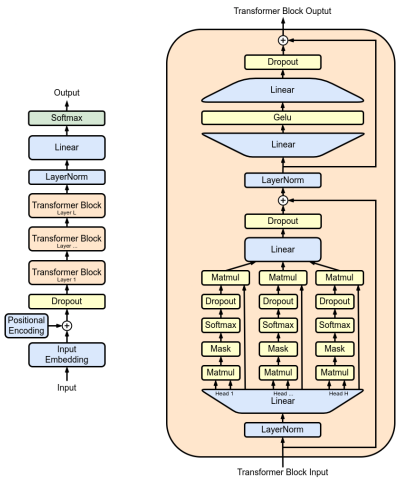


Fig. 2. GPT-2 Transformer Model Diagram

dropout at the last fully connected classification layer and weight decay during training to further reduce the effects of overfitting. A breakdown of the entire model architecture is provided in Appendix C.

This model was then trained with the same 90%/10% train/validation split and 101 testing examples we withheld for final testing as done with the DistilBERT text classification model. The hyperparameters for this model were then tuned using SigOpt for maximum validation accuracy. The hyperparameters we chose to tune were the weight decay and learning rate. We also tuned the dropout rate at the classification layer manually.

Similar to the DistilBERT model methods, we re-trained our new GPT-2 text classifier model and tuned the hyperparameters and dropout rate for a variety of train/validation splits from 90%/10% down to 10%/90%. These results were then compared with the DistilBERT Transformer based text classifier.

*3) Naive Bayes Article Classifier:* For the third and final supervised model we implement a Naive Bayes classifier. Text was pre-processed by removal of digits, removal of the word "advertisement", removal of stopwords, removal of text within quotes, removal of words such as "associated press", splitting up contractions, lemmatization, and tokenization. Lists of categorized "biased terms" per category from both data sets was pre-processed in a similar fashion. These "biased" phrases deviate from a neutral center, rather than towards only right or left. Python library `sklearn`'s text vectorizer tokenizes text into inidividual words or phrases unique to a vocabulary corpus given as input. Three unique corpi were used to vectorize the text; a standard English word-for-word corpus; a corpus that explicitly only counts tokens that appear within the lists "biased word and phrases"; and a corpus that excluded "biased words and phrases'. Using the function `CategoricalNB` three models from these three corpi were constructed for nine news categories denoted by ['tech & finance', 'us news', 'human rights', 'environment', 'us immigration', 'military & intelligence', 'sports', 'education', 'world news']. A workflow can be seen in *Figure 3*.

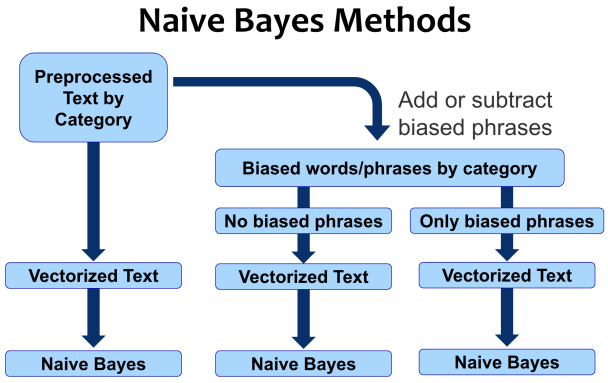


Fig. 3. Naive Bayes Method Workflow to Give Three Models for Each News Category

### C. Unsupervised Article Classification

*1) OpenAI Chat API Classifier:* We conduct zero-shot learning on OpenAI's powerful Chat API. Among GPT-3.5 models, we choose `gpt-3.5-turbo` because it is the most capable and has the lowest cost. The OpenAI models are all paid service, with a price proportional to number of tokens used. For financial feasibility, we estimated the total cost of classifying the whole dataset, using OpenAI's `tiktoken` Python library. The calculation result shows the cost has order of the magnitude of $0.1, which is feasible for our team.

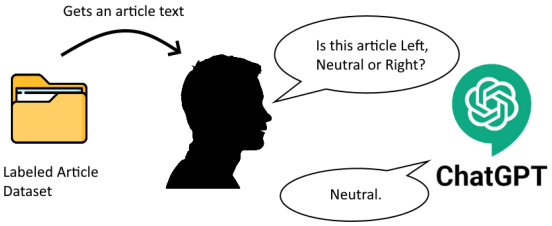


Fig. 4. A Brief Summary of Unsupervised Chat API Tactic

For each article (or sentence) in our labeled dataset, we construct a prompt containing the article's text and sent the prompt to OpenAI's Chat API. We then record its response and compare it to the ground truth.

In terms of prompt engineering, we mainly focus on the following aspects:

- System message: This is the starting message of a conversation, and it helps set the behavior of the assistant. To tailor it to the needs of the task, we set it as `"You are a helpful assistant that detects media bias."` However, according to documentation, `gpt-3.5-turbo-0301` does not always pay strong attention to system messages, and future models will be trained to pay stronger attention to system messages.
- User message: Just like in ChatGPT, the user messages help instruct the assistant and are the most important elements in prompting. For our task, we use `'Classify the text from news media into one of three labels, "Left", "Neutral", or "Right" in terms of US politics.'` This instruction plainly states our task, and is intuitive and straightforward. We include some domain and context, such as "from news media" and "US politics" in the instruction, to make the prompt more concrete and more aligned with our task.
- Prompt structure: We use the following structure in the following format [11]:

TABLE I
PROMPT STRUCTURE OF ZERO-SHOT CLASSIFICATION

| | |
|---|---|
| *Description* | `Classify the text from news media into ... \n` |
| *Input indicator* | `Text:\n` |
| *Input* | `YouTube is making clear there ... \n` |
| *Output indicator* | `Label:\n` |

Specifically, we need the indicators (delimiters) and newline symbols because we need to create clear boundaries between instructions (metalanguage) and texts (language).

- Answer format: For this classification task, we want the model to reply in only one word from the three labels, namely "Left", "Neutral" or "Right". We want to enforce this answer format and avoid any unnecessary or undefined replies including rejections like "none" and synonyms like "central". For this purpose, we add more instructions emphasizing the answer format: `"It is crucial that you answer in only one word. Don't use any word other than the three labels."`

Besides prompt engineering, we also tuned the chat model by modifying two key hyperparameters:

- `temperature`: High values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We used a lower temperature since our task is classification, not creative writing,
- `max_tokens`: This is for limiting response to a certain length. Since we only expect three possible class labels as responses from the API, and according to `Tokenizer`, each of them only consists of 1 token, we can safely set the limit to a low number like 1. This measure can potentially save the total number of tokens used and thus reduce the cost.

## IV. Experiments and Results

In this section, we present the results from the methods described in detail previously and analyze our findings related to the goals for this project.

### A. Supervised Article Classification

After training a DistilBERT and GPT-2 Transformer based text classifier for political bias detection, we were able to achieve the following results for test accuracy vs. train/validation splits shown in *Figure 5*.

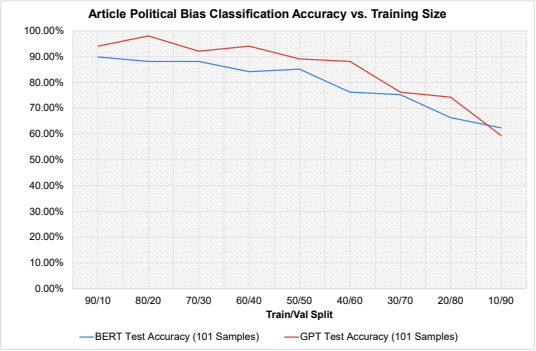


Fig. 5. BERT and GPT-2 Transformer Model Classification Results

These results show a significant increase in test accuracy for the DistilBERT model vs. the original BERT model results achieved by Simoes and Castanos from Stanford University [7]. While the original BERT model produced test accuracy results of approximately 66% on our dataset, our new "lighter" DistilBERT model achieved a maximum test accuracy of approximately 90%. These very high test accuracy results were very surprising but not unreasonable as BERT text classifiers have been known to achieve 90%+ test accuracy results on other text classification tasks. Another explanation for this result could come from the curation of our MBIC dataset. The articles chosen to be classified as left or right biased very chosen specifically to be *very* left or *very* right

biased for an extremely large political bias gap between each of the classes (left, center, right). This could have helped our classifier reach such a high test accuracy especially with a much "lighter" and easier to train model. However, significant further analysis would be required to better understand the features used from the articles to achieve such high test accuracy and validate the results we achieved.

When comparing the DistilBERT and GPT-2 Transformer based text classifier results, we see that GPT-2 was able to achieve slightly better test accuracy for classification for almost all the train/validation splits. This is a huge breakthrough in the world of large text classification using Machine Learning as the GPT-2 architecture (traditionally used for text generation) is able to not only match Dis tilBERT (state of the art for text classification) accuracy but also exceed it. With a 80%/20% train/validation split, the GPT-2 model was able to achieve an astounding 98% test accuracy (99/101 correctly classified). The confusion matrix for this classification result is also provided in *Figure 6* along with the precision, recall, and F1-scores for each class in *Table II*. This result certainly raises concerns for the validity of the methods used to achieve this test accuracy. To perform our due diligence, we searched for any leaking of test data into our training datasets, "lucky" training results, and leaking of ground truth labels into training article text. We were unable to find any major issues with the methods used and present these results with an asterisk. Significant further analysis would be required to better understand the features used from the articles to achieve such high test accuracy and validate the results we achieved.

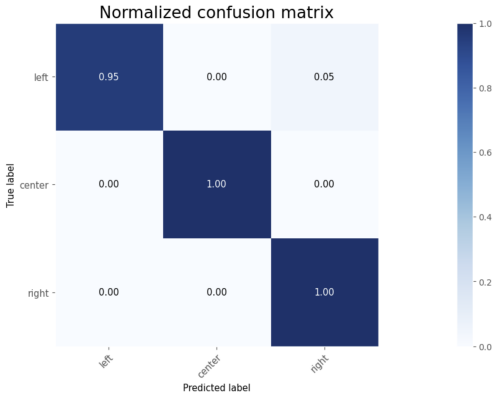


Fig. 6. GPT-2 80%/20% Classification Confusion Matrix Results

TABLE II
GPT-2 80%/20% CLASSIFICATION PRECISION, RECALL, AND F1-SCORE RESULTS

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| left | 1.00 | 0.95 | 0.98 | 44 |
| center | 1.00 | 1.00 | 1.00 | 12 |
| right | 0.96 | 1.00 | 0.98 | 45 |
| | | | | |
| accuracy | | | 0.98 | 101 |
| macro avg | 0.99 | 0.98 | 0.99 | 101 |
| weighted avg | 0.98 | 0.98 | 0.98 | 101 |

*1) Naive Bayes Article Classifier:* Comparing accuracy across all nine categories we were able to discern three main regimes, as seen in *Figure 7*. Regime one is seen in the 'sports' category where all three vocabulary corpi give a similar, low accuracy. Sports are generally an unbiased news topic and it is expected that predicting bias is little better than a random guess accuracy of 33%. The second regime is seen in 'US news', where corpi either excluding or solely including biased phrases perform with lower accuracy in comparison to full word-for-word vectorized text. This supports the idea of an instance where full context carries the weight of overall bias in an article. The third and final regime can be seen in category 'military & intelligence'; here the highest return on accuracy is achieved when only considering biased words/phrases. This suggests that there are clear key terms that both bias extremes use regularly within this category.

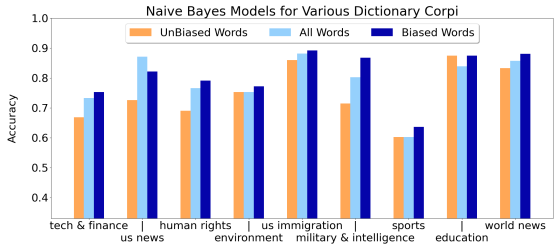


Fig. 7. Naive Bayes Test Accuracy for Nine General News Categories

## B. Unsupervised Article Classifier

*1) OpenAI Chat API Classifier:* Over the entire `MBIC` dataset, OpenAPI's ChatGPT achieved an accuracy of 46.8%. The confusion matrix is shown in *Figure 8*. We can observe that the classifier performs fairly well on "left" and "center" labeled data, but perform poorly on "right" labeled data. It is likely due to the discrepancy between different interpretations of the labels or the inherent human bias of the annotations in the dataset.

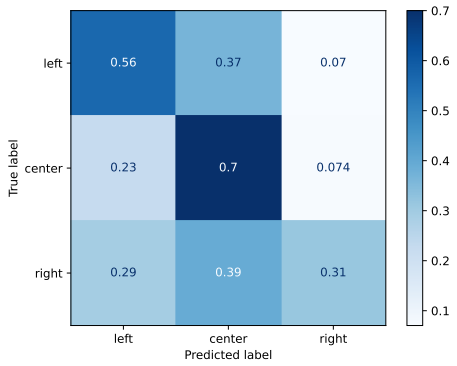


Fig. 8. ChatGPT Classification Confusion Matrix Results

## V. Discussion and Conclusions

Notably, all of our methods of classifying media bias struggled to indentify and correctly classify satire. ChatGPT tended to take imput texts very literally, and struggled to understand sarcastic or joking tones. In addition, while language models may be able to understand the context and implications of certain phrases within sentences, they struggle to understand the ever-changing landscape of the world as a whole.


Fig. 9. Twitter Post Mocking Donald Trump Suggesting Virginians Need Guns to "Guard Potatoes"

For example, the tweet shown in *Figure 9* would require knowledge of the context of the current events of the time of the tweet in order to properly classify its political bias. Current events are often difficult for LLMs such at BERT or ChatGPT to understand because of the time that goes into developing, collecting data for, and training these models.

Upon deeper inspection of ChatGPT's responses to article classification, we discovered that ChatGPT focuses on the tone of the article as well as its sources. If ChatGPT doesn't see a strong emotional tone or questionable sources, it will view the article as neutral. ChatGPT also looks if the author took a stance or promoted/criticized the topic of the article. It makes sense for ChatGPT to play it safe if it is unable to detect any clues. While ChatGPT sometimes feels like it can make emotional decisions through its massive database, it is still an AI that produces output based on its known facts.

Another possible explanation for the results could be the fact that we weren't able to give ChatGPT the full article. Because the API has a max limit of 4096 tokens per prompt, we only gave ChatGPT the first 4000 characters of the article. This means that ChatGPT could miss out on some context at the end of an article to make a more informed decision.

Also, it's worth noting that the datasets we used are inherently biased due to specific demographics of the annotators. In comparison, ChatGPT is trained on millions of web pages collected from the Internet [12] and therefore generally "interpret" the labels in a much different way. Always treating the annotations as ground-truths may not be the most scientific way to guide our model. The root cause of this problem is our ambiguous usage of the terms and the lack of a clear definition.

Overall, ChatGPT is still more accurate than random guessing, and it can still be used to classify a single article. Other users can also obtain an explation from ChatGPT instead of a single world response to get a more informed reasoning behind ChatGPT's choice.

## VI. Future Work

For our supervised training results, the main area of future work revolves around verifying our test accuracy results for both the DistilBERT and GPT-2 models. We would like to make sure these results are also supported with a larger test dataset with more variety in test data. One of the areas we would like to evaluate are the variance of our dataset in terms of how "left" or "right" our articles were. We are surprised by the high accuracy results we achieved and would like to make sure these results hold on more article data that is possibly less polarized.

Another area of future work we would like to pursue revolves around retraining our supervised models and injecting bias into our training datasets for a robustness evaluation. The main goal of this project was to create a media bias article classifier that is robust to human bias when labeling data for training. To evaluate this, we present a plan to shift training data classification labels either "left" or "right" for a certain percentage of training data (shift percentages) to retrain our models . These models will then be used to classify our un-modified test data for accuracy evaluation. This process will be repeated for multiple train/validation splits and shift percentages to get a strong understanding of the effect training labels have on our model generation after the text is embedded using pre-trained DistilBERT and GPT-2 transformer models. The goal for this procedure is to identify

the best train/validation splits for a strong classifier that is minimally affected by various levels of bias (shift percentages) in the training data labels.

Another expansion of this project revolves around the fact that news is not discrete in terms of political bias. To account for this, we would like to expand this project to classify text on a spectrum from "left" to "right" rather than three discrete categories (left, center, right). This would require acquiring a dataset that is labeled for political bias on a spectrum further complicating the requirement that training data is kept as unbiased and accurate as possible. If this dataset could be procured or acquired, we would be able to greatly improve our ability to analyze text with both an identification of political bias and intensity.

For our Naive Bayes approach, we were able to achieve some promising results given this very simple model. Future work would revolve around tuning the hyperparameters used for test accuracy improvements. Specifically, the usage of keywords and non-keywords can be tuned to capture more or less details (number of words used in classifier). This could help us increase or decrease the complexity of our classifier for more specific or more general classification. Over-tuning in either direction would lead to overfitting or underfitting but optimal tuning would allow us to create a very powerful model using simple classical methods for maximized classification accuracy.

Regarding our ChatGPT API Classifier, one potential area of future work is to fine-tune our model for a GPT-3.5-turbo or GPT-4 model (waitlist to access API). OpenAI's API currently has options to fine-tune some of their models given a training file and boasted significant classification improvements with minimal training data. Fine-tuning will allow the model to get better results than high-quality prompt design and decrease latency. There were a few reasons why we didn't pursue this option for the project. One reason was because this classifier would go from unsupervised to supervised learning due to it needing training data. Another reason was due to costs, as training and using training a fine-tuned model is significantly more expensive than using the basic ChatGPT API, especially for a GPT-3.5-turbo or GPT-4 model.

Overall, the results from our project certainly produced a lot of important results that can be used to further work on research for stronger article bias text classification methods.

## VII. Acknowledgements

We would like to thank Prof. Honglak Lee, GSI Jongwook Choi, GSI Yunseok Jang, and GSI Anthony Liu for a great learning experience in this course and all the assistance we received on this project.

## REFERENCES


[1] F. Hamborg, K. Donnay, and B. Gipp, "Automated identification of media bias in news articles: an interdisciplinary literature review," *International Journal on Digital Libraries*, vol. 20, pp. 391–415, 12 2019.

[2] L. Atkins, *Skewed : a critical thinker's guide to media bias*. Amherst, New York: Prometheus Books, 2016.

[3] W. P. Eveland and D. V. Shah, "The impact of individual and interpersonal factors on perceived news media bias," *Political Psychology*, vol. 24, pp. 101–117, 3 2003.

[4] T. Spinde, "Mbic a media bias annotation dataset." https://www.kaggle.com/datasets/timospinde/mbic-a-media-bias-annotation-dataset, Jan 2021.

[5] "Huggingface bert-base-uncased." https://huggingface.co/bert-base-uncased.

[6] "Huggingface gpt2." https://huggingface.co/gpt2.

[7] A. A. Simoes and M. Castanos, "Fine-tuned bert for the detection of political ideology," in *Fine-Tuned BERT for the Detection of Political Ideology*, 2020.

[8] "Newspaper3k: Article scraping & curation - newspaper 0.0.2 documentation."

[9] J. P. Usuga-Cadavid, S. Lamouri, B. Grabot, and A. Fortin, "Using deep learning to value free-form text data for predictive maintenance," *International Journal of Production Research*, vol. 60, no. 14, pp. 4548–4575, 2022.

[10] "Huggingface bert-base-uncased." https://huggingface.co/distilbert-base-uncased.

[11] P. Daniel, "Prompt engineering – part ii – how to construct prompts," Feb 2023.

[12] E. Ferrara, "Should chatgpt be biased? challenges and risks of bias in large language models," *arXiv preprint arXiv:2304.03738*, 2023.

## APPENDIX A
## EXAMPLE RAW TRUNCATED ARTICLES FROM MBIC DATASET

| Text | Label |
|---|---|
| Senate Majority Leader Mitch McConnell (R-KY) said on the Senate floor on Monday that some Democrats are "embarrassed" that House Speaker Nancy Pelosi (D-CA) and Senate Minority Leader Chuck Schumer (D-NY) blocked the bipartisan coronavirus relief package. McConnell spoke on the Senate floor after Democrats blocked a bipartisan package that would alleviate the economic effects of the coronavirus outbreak. Four swing state Senate Democrats -- Sens. Jeanne Shaheen (D-NH), Doug Jones (D-AL), Tina Smith (D-MN), and Gary Peters (D-MI) -- voted against the bipartisan package to provide Americans relief. The Senate Republican leader said that some Democrats told him in private they were embarrassed by Pelosi and Schumer's political stunt. McConnell said Democrats "ought to be embarrassed." He added, "In fact, I've heard from some who are embarrassed, talking like this is not some juicy political opportunity." "This is a national emergency," he added. McConnell asked rhetorically, "Why are Democrats filibustering the bipartisan bill they helped write?" The Senate majority leader chastised Democrats for pushing their "wish list" of special interest provisions to be included in the bill. | right |
| U.S. President Donald Trump wanted to authorize shooting Central American migrants in the legs and building snake- and alligator-infested moats to stop them from entering the United States. Instead, his administration continues to house them in overcrowded detention camps on the southern border. United States congresswoman Alexandria Ocasio-Cortez condemned the Trump administration for running "concentration camps" earlier this year. Though she was by no means the first to describe migrant detention centres as concentration camps -- disgraced Arizona Sheriff Joe Arpaio boasted his "tent city" in the Sonora Desert was just that -- her comments ignited a firestorm of controversy. Republican congresswoman Liz Cheney complained concentration camp comparisons "demeaned the memory" of six million Jews murdered during the Holocaust and she urged AOC to learn some "actual history." | left |
| "FIFA bans former referee for life for bribery, match-fixing. One of soccer's most infamous match-fixing cases was settled Thursday when a referee notorious for corrupt calls was banned for life. The corrupt games in Ibrahim Chaibou's career were key to revealing how easily international friendlies could be manipulated for betting scams, forced FIFA to change the rules for appointing referees, and helped expose the influence of convicted fixer Wilson Perumal. "Chaibou was probably the most corrupt referee the game of football has seen" former FIFA investigator Chris Eaton told The Associated Press on Thursday. Still, it took more than eight years to confirm his life ban from any involvement in soccer. FIFA ethics committee judges found the referee from Niger guilty of taking bribes to corrupt international friendly games in 2010 and 2011, soccer's world governing body said. Chaibou was fined 200,000 Swiss francs ($201,000), though it is unclear what power FIFA has to make the long-retired referee pay. He can appeal to the Court of Arbitration for Sport." | center |

## APPENDIX B
## DISTILBERT TEXT CLASSIFICATION MODEL ARCHITECTURE

```
DistilBertForSequenceClassification(
  (distilbert): DistilBertModel(
    (embeddings): Embeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (transformer): Transformer(
      (layer): ModuleList(
        (0-5): 6 x TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
  )
  (pre_classifier): Linear(in_features=768, out_features=768, bias=True)
  (classifier): Linear(in_features=768, out_features=3, bias=True)
  (dropout): Dropout(p=0.2, inplace=False)
)
```

## APPENDIX C
## GPT-2 TEXT CLASSIFICATION MODEL ARCHITECTURE

```
GPT2ForSequenceClassification(
  (transformer): GPT2Model(
    (wte): Embedding(50257, 768)
    (wpe): Embedding(1024, 768)
    (drop): Dropout(p=0.1, inplace=False)
    (h): ModuleList(
      (0-11): 12 x GPT2Block(
        (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (attn): GPT2Attention(
          (c_attn): Conv1D()
          (c_proj): Conv1D()
          (attn_dropout): Dropout(p=0.1, inplace=False)
          (resid_dropout): Dropout(p=0.1, inplace=False)
        )
        (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (mlp): GPT2MLP(
          (c_fc): Conv1D()
          (c_proj): Conv1D()
          (act): NewGELUActivation()
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
    (ln_f): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
  )
  (pre_classifier): Linear(in_features=768, out_features=768, bias=True)
  (classifier): Linear(in_features=768, out_features=3, bias=True)
  (dropout): Dropout(p=0.2, inplace=False)
)
```

## APPENDIX D
## AUTHOR CONTRIBUTIONS

| Task | Gabby Jones | Jessica Ma | Shivam Patel | Wesley Tsai | Tao Zhou |
|---|---|---|---|---|---|
| Abstract | ✓ | | | | |
| Introduction | ✓ | | | | |
| Related Work | | ✓ | | | |
| Dataset Pre-Processing | | ✓ | ✓ | | |
| BERT Classifier | | | ✓ | | |
| GPT-2 Classifier | | | ✓ | | |
| Naive Bayes Classifier | ✓ | ✓ | | | |
| OpenAI Chat API | | | | ✓ | ✓ |
| Discussion/Conclusions | | | | | ✓ |
| Future Work | | | ✓ | ✓ | |
| Appendices | | | ✓ | | ✓ |