HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 2 (2024)

Bias Detection in Media: An NLP-Based Approach using Corpus Statistics and Sentence Embeddings

Neeraj Gummalam[1] and Clayton Greenberg[#]

[1]Henry M. Gunn High School, USA
[#]Advisor

ABSTRACT

In this paper, we implement a Natural Language Processing (NLP) solution for binary classification to categorize a sentence as biased or unbiased. Detecting bias is a challenge in the media today, but can be utilized to help readers identify which sources portray bias. The general approach to classifying a sentence as biased or unbiased involves representing words and sentences using probability or pretrained vectorization models. Our final model only contained probabilistic data about the connection between words, sentences, and each class. We used Pointwise Mutual Information (PMI) and Term Frequency Inverse Document Frequency (TF-IDF) as heuristics for finding the relationship between sentences and the biased and unbiased classes. We also leveraged Google's Universal Sentence Encodings (USE) to capture the meaning of the sentences. Our results revealed a possible limitation in USE's training data in terms of bias detection. Through topic analysis, we were able to uncover insights surrounding which topics are characterized by minimal bias. We were able to use these discoveries to contextualize the model's performance.

Introduction

In today's rapidly evolving media landscape, the power of information dissemination is unmatched. Yet, the media's influence is often overshadowed by bias. Embedded within news stories, this bias can subtly alter narratives and shape public understanding. These sources of information that we digest on a regular basis can exacerbate political polarization by fueling existing ideological "echo chambers" with bias (Bail et al., 2018). As we navigate an interconnected world of information sharing, understanding the nuanced presence of bias in the media becomes vital. This paper examines the intricacies of media bias, its effects, and potential remedies.

We aim to explore and evaluate various strategies for detecting bias using NLP, and discover strengths and limitations of each. Through NLP techniques and binary classification modeling, our objective is to accurately label sentences as either "biased" or "unbiased". The desired outcome of this research is a comprehensive and reliable system capable of processing individual sentences and discerning their underlying bias. With this system, users will receive a clear and concise output indicating whether a sentence is biased or unbiased, aiding in the identification and understanding of biased sources.

There have been several approaches to identifying bias in the media. One approach used TF-IDF and sentence embeddings and found that sentence embeddings performed better for detecting bias (Nadeem and Raza). We found an enhanced implementation of TF-IDF for bias detection, which allowed it to outperform the sentence embeddings. We also tried using a vectorizer, USE, to generate sentence embeddings. USE is a powerful tool that represents a sentence as an embedding that captures the semantic meaning and contextual information of the input sentences (Cer et al., 2018). Another approach used neural networks and preserved word ordering to infer bias in a source (Chao et al., 2022). However, we wanted to follow a simpler approach.

We also performed topic analysis to explore which topics contain the most biased text corpus and hypothesized potential reasons for such findings.

Methods

Dataset

This work uses the Media Bias Including Characteristics (MBIC) dataset (Spinde et al., 2021). In this dataset, there is a diverse collection of excerpts from different news sources, which are all shown in Figure 1, and authors. From this, we were able to identify which news sources had a higher biased to unbiased ratio in this dataset.

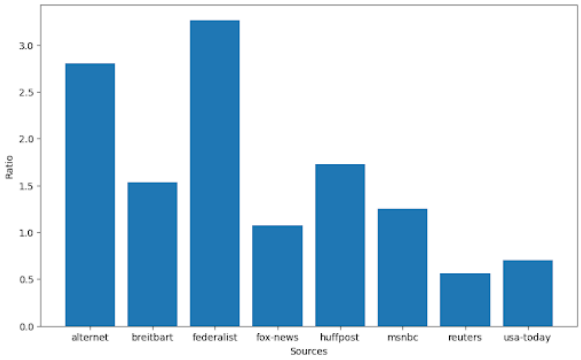

Figure 1. Ratio of bias to non bias in news sources. Each bar shows a news sources' ratio of biased to unbiased sentences in the MBIC dataset.

Several features included were about the writer's demographics as well, such as age, gender, higher education, and background in the English language. After performing some exploratory data analysis, we found a small but significant positive correlation between the writer's age and the percentage of biased sentences (Pearson $r = 0.28$, $p < 0.05$). The scatterplot shown in Figure 2 illustrates this correlation.

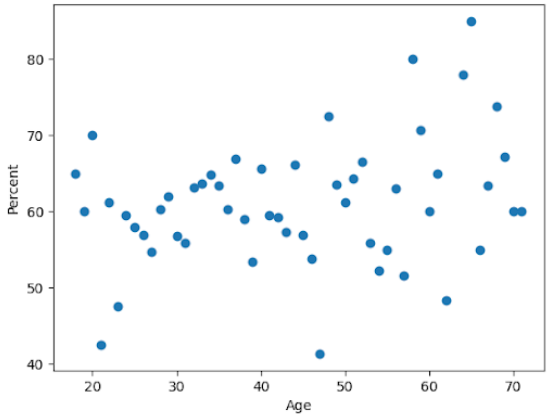

Figure 2. Percent of biased sentences by writer's age. Each point shows what percent of sentences (y) are biased for writers with a certain age (x).

This information can be used if the name of the news outlet is picked up when scraping. However, the core objective of this paper is to identify bias from within the sentence itself, without informative metadata that may provide context.

This dataset contains over 17k sentences. Each record contains a raw sentence from a news article. In addition to each sentence, there is a label that classifies the sentence as "biased" or "unbiased". Approximately 60% of the sentences in this dataset are labeled as biased. Therefore, the dataset is not too severely imbalanced.

Accompanying each sentence is its topic. There are a total of fourteen unique topics, and each sentence is represented by exactly one topic. The "topic" feature is useful for data exploration, as well as for topic analysis. In fact, we used these topics to understand how the ability to detect bias is different across different topics. We decided against using the topics in training the model because then we would have to classify the topic of a novel sentence. This represents a uniquely challenging research direction of its own and remains outside the scope of this work.

Before training our models, we applied a preprocessing procedure on the text. After splitting the sentence into an array of words, we removed all stop words and all punctuation. This converted each sentence into an array of content-bearing tokens. From there, we concatenated all of these words into a string again for use with Transformer models that expect raw text.

We used 90% of the data for training, and the remaining 10% for testing. This breakdown allowed us to create more accurate models by providing a large amount of training data while still retraining an adequate amount of data for testing.

Model Implementation

The first step we took to classifying a sentence was to identify the words that are the most discriminatory, or the words that are the most biased. Our initial idea was to count the frequency of each word in each class, as shown in Table 1.

Table 1. Preview of words with the greatest difference between frequency in each class.

Word	Biased Counts	Nonbiased Counts	Difference	Direction
trump	3288	1674	1614	biased
white	1040	496	544	biased
president	1517	979	538	biased
people	971	537	434	biased
democrats	772	391	381	biased

We used CountVectorizer from the scikit-learn library to perform word count statistics. From there, we fed the vectors into a Random Forest Classifier. By performing a hyperparameter grid search, we found the most optimal range of discriminative words to use as the vocabulary to be approximately 3000. However, our accuracy score was below 70%. This is because our first method used to find discriminative words did not consider the global frequency of each word or the class imbalance. This was a problem because words with the largest difference might not be more biased than other words: they might just appear more frequently in both corpora. Using a ratio of biased counts to unbiased counts as a medium instead of difference does not fully address the issue because it does not take the imbalance of biased to unbiased sentences in our dataset into account.

To address this problem by way of feature engineering, we decided to find the Pointwise Mutual Information (PMI) value for each word in our vocabulary, for each class. To find the biased and unbiased PMI

values, we used the equation, $PMI(w,c) = log_2 \frac{P(w,c)}{P(w) \cdot P(c)}$ where "w" represents any word that appears in both classes and "c" represents the class, either biased or unbiased. This method of calculation gave us features which yielded better performance because it quantifies how associated a word is with a certain class using probability. We found that the unbiased PMI value was consistently higher than the biased PMI value. This can also show that the type of words associated with the biased class is more one-dimensional in comparison to the unbiased class. We used word-clouds to find words most associated with each class to test this hypothesis.

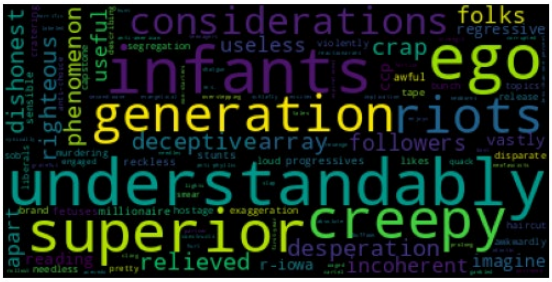

Figure 3. The Biased Class Word Cloud. Larger words represent words with a higher biased PMI value, implying that these words have stronger association with being used in biased sentences.

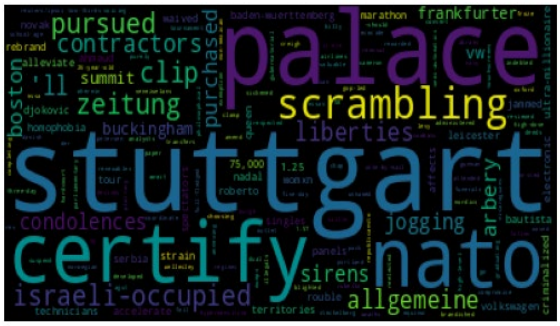

Figure 4. The Unbiased Class Word Cloud. Larger words represent words with a higher unbiased PMI value, implying that these words have stronger association with being used in unbiased sentences.

As shown in the word clouds, the words associated with the biased class, shown in Figure 3, are more negative and descriptive, whereas there is a larger variety of the type of words in the unbiased class, shown in Figure 4. The words here are a completely different set of what the biased words by count show, and they seem much more associated with bias than the earlier set of words.

Now that we have how associated a word is with a class, we can find how associated a sentence is with a class by simply finding the sum of all PMI values of the words contained in a sentence, for each class. Our preliminary model consisted of only these two PMI values of the sentence. We implemented two classifiers: a Support Vector Machine and a Random Forest. It was found that the SVM outperformed the Random Forest in terms of accuracy.

Next, we explored vector representations. We used Universal Sentence Encodings (USE), which are vector representations of sentences that capture their semantic meaning. These encodings are obtained from Google's pre-trained deep neural network models They enable the transformation of each sentence into a fixed-length vector that encodes its semantic content. By using these encodings, we aimed to capture deeper information about the sentences, allowing the model to make better predictions.

Another vector that we explored was TF-IDF. Similarly to our approach with USE, we combined the TF-IDF vector with the two PMI values for each sentence. However, this feature engineering approach did not yield promising results either.

We decided to try a dual TF-IDF model, where we used two TF-IDF vectors: a biased TF-IDF, and an unbiased one. To create a TF-IDF vector for each class, we defined the corpus used to be the biased or unbiased corpus. This improved the performance of the model because the features contained information on how important words in sentences are for each class. Although similar, this is different from PMI because PMI does not take the probability of a word in a sentence into account. PMI only takes in word and class probability in the corpus. We decided to combine these and our two PMI values to create a final vector which would be used for classification. This resulted in our final model which yielded peak performance.

Results and Discussion

Using the USE vector alone, we observed that the Random Forest algorithm outperformed the Support Vector Machine (SVM) by a margin of 2 percentage points in terms of accuracy. This suggests that the USE vector on its own is more favorable for creating multiple decision trees within the Random Forest framework. However, our results took a significant turn when we incorporated the two PMI values at the end of the USE vector and conducted tests using both the Random Forest and SVM. Surprisingly, the SVM demonstrated superior performance, indicating that the addition of the PMI values enhanced the SVM's ability to establish a hyperplane that effectively separates the two classes, i.e., biased and unbiased sentences.

Our singular TF-IDF approach was the least effective. TF-IDF calculates the importance of a word in relation to both the sentence and the entire corpus. While TF-IDF can be useful for identifying key words and topics within a sentence or document, it does not provide insight related to bias or unbias. Our dual TF-IDF approach performed better, because it calculated the weight of a word in relation to the sentence and also the biased and unbiased class. The dual TF-IDF approach ended up being our most successful and final model throughout our research.

To evaluate the performance of our models, we calculated the precision and recall. Precision represents the model's accuracy in classifying sentences as biased, while recall indicates the model's ability to predict bias when it is present. From there, we calculated the F-score for our models to be a final measure of how well it performed, taking into account both precision and recall.

The results in Figure 5 show the precision, recall, and F-score of our models.

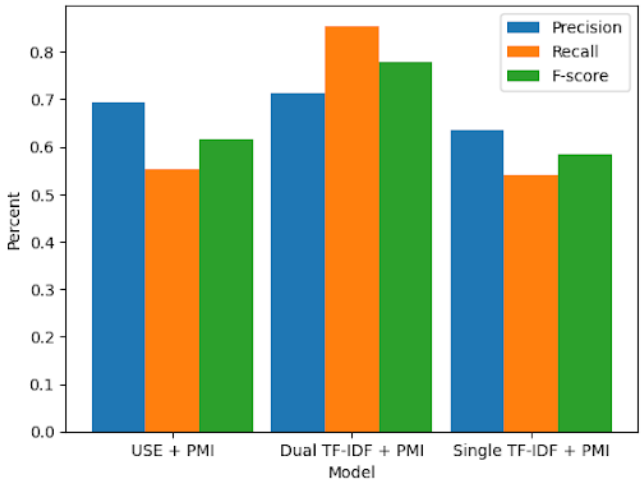

Figure 5. Performance by Input Data. The performance of each model, described by input data, is represented by its precision (blue), recall (orange), and F-score (green).

The recall for our USE and PMI model was relatively low at 55.3%, suggesting that the model frequently failed to identify instances of bias. One hypothesis which could explain the low recall score is that there was not a sufficient amount of biased text used for training the Universal Sentence Encoder model.

The recall for our dual TF-IDF model was 30.1 percentage points higher than the aforementioned model, meaning that this model performed significantly better in detecting bias. In fact, this was our best model, resulting in an overall F-score of 77.7%

In summary, our study revealed that the single TF-IDF performed poorly in detecting bias, as it primarily focuses on the relevance of words within a sentence and the corpus as a whole, rather than their specific bias-related properties. The dual TF-IDF and PMI methods performed much better because they include valuable information on the relation between sentences and each class. Our proposed rationale for why the USE had such low scores is that there was not a sufficient amount of bias in the training data.

Topic Analysis

Because our dataset contained helpful information about the topic of each sentence, we decided to explore the variation in precision, recall, and F-score across each topic to see if certain topics are harder to detect bias in.

Our dataset assigned one of 14 different manually identified topics to each sentence. We analyzed our dataset in terms of these 14 topics. We wanted to see if there was variation in the ability to predict bias throughout the topics.

After evaluating the precisions, recalls, and F-scores of each topic, we found that the precision and recall varied by topic, as seen in Figures 6 and 7.

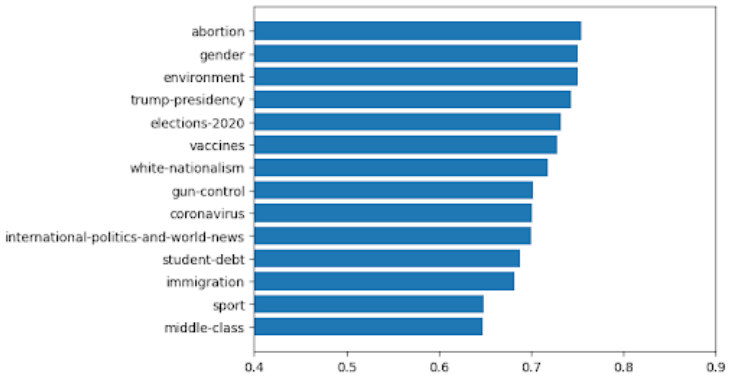

Figure 6. Precision of Topics. Each bar represents the precision of our model for each topic.

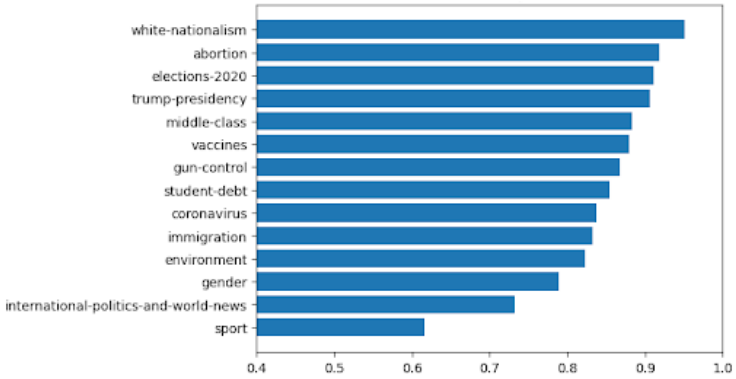

Figure 7. Recall of Topics. Each bar represents the recall of our model for each topic.

These results show that certain topics are harder to classify as biased or unbiased, and certain topics are easier. For example, the precision and recall for sports is 64.8% and 61.6%, respectively, whereas the precision and recall for abortion is 75.4% and 91.8%, respectively.

We found that only 40.7% of sentences regarding sports were biased, among the lowest of all topics in the dataset. In the entire dataset, the amount of sentences that were biased was around 60%. To find out if the difference in bias proportion was the cause of a low precision and recall, we determined whether our model predicts a similar proportion of bias to the true bias proportion in the topic.

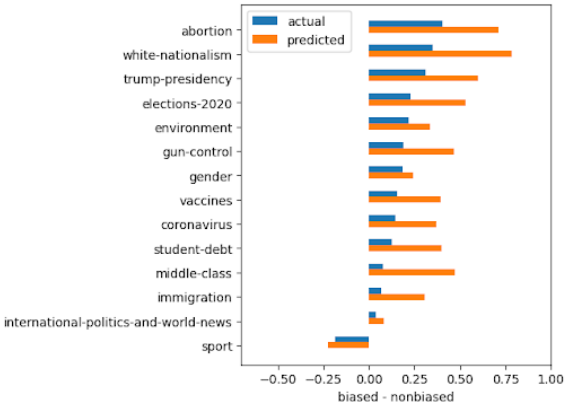

Figure 8. Difference of Bias and Unbias for each topic. The orange bar represents the difference of predictions by our model, and the blue bar represents the actual difference in the dataset.

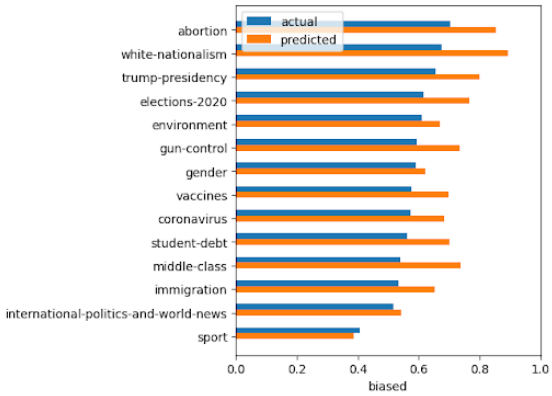

Figure 9. Proportion of Bias to Unbias for each topic. The orange bar represents the proportion of predictions by our model, and the blue bar represents the actual proportions from the dataset.

As shown in Figures 8 and 9, our model predicted a similar amount of bias for the topic of sports. In fact, it seemed to be more accurate than all of the other topics. To explore the most significant words in each topic, we used PMI and generated word clouds, as seen in Figures 10 and 11.

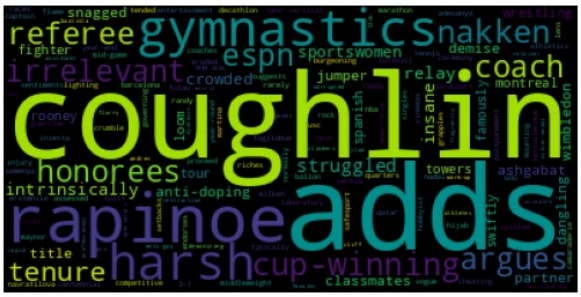

Figure 10. Sports Topic Word Cloud. The larger words are words that have a higher sports PMI, and therefore have higher association with this topic.

Figure 11. Abortion Topic Word Cloud. The larger words are words that have a higher abortion PMI, and therefore have a higher association with this topic.

Figure 10, the sports word cloud, shows many names and factual information in comparison to Figure 11, the abortion word cloud. From this, we concluded that sports is a more factual, lower bias topic, whereas many other topics are more subjective and morality-based.

Conclusion

We tried implementing several feature engineering approaches and multiple models to classify a sentence as biased and unbiased. Our first strategy was to find how related a word is to a certain class. Our findings show that a very effective way to find how biased a word is using the PMI equation, which measures how important a word is with respect to the frequency of the word, the class, and the word in the class.

The next strategy we pursued was employing vectorizers to represent a sentence as a vector. We found that TF-IDF, when used individually, performed poorly because it doesn't capture any type of meaning or how important a sentence is in relation to bias. We found another way to find how important a sentence is to each class using a "dual TF-IDF" strategy. Essentially, we created a biased and unbiased TF-IDF vector by splitting our large corpus into two different corpora, with one corpora for each class. Once combined with PMI, this model proved to be the most effective at the classification task, giving us an overall F-score of 77%. After analyzing the topics, we found that certain topics, such as abortion, are far easier to detect bias in. We also learned that sports is a topic that is harder to detect bias in because it is a topic that contains a higher proportion of unbiased sentences in this dataset as compared to other topics.

Another vector we tested was Universal Sentence Encodings. Although it does a good job at capturing semantics, the domain mismatch may have resulted in limitations that made it harder to detect bias. Our proposed explanation was that there was a lack of bias in the USE training data, which hindered its ability to capture bias-related information of a sentence.

Overall, the highest performing model from our experiments can be used to create helpful tools to help readers identify bias when reading articles. Bias in the media is a large problem in today's day and age, and this research can be utilized to counter the problem or, at the very least, give readers context when they consume media.

Acknowledgments

Thank you to Sejal Dua for her feedback and support in this paper.

References

Bail, C.A.; Argyle, L.P.; Brown, T.W.; Bumpus, J.P.; Chen, H.; Hunzaker, M.F.; Lee, J.; Mann, M.; Merhout, F.; Volfovsky, A. Exposure to opposing views on social media can increase political polarization. Proceedings of the National Academy of Sciences of the United States of America 115(37), 9216–9221 (2018). https://www.doi.org/10.1073/pnas.1804840115

Nadeem, M.U.; Raza, S. "Detecting Bias in News Articles using NLP Models," Stanford CS224N Custom Project, Stanford University. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1224/reports/custom_116661041.pdf

Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; St. John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Strope, B.; Kurzweil, R. Universal Sentence Encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 169–174, Brussels, Belgium. Association for Computational Linguistics (2018). https://doi.org/10.48550/arXiv.1803.11175

Chao, Z.; Molitor, D.; Needell, D.; Porter, M.A. "Inference of Media Bias and Content Quality Using Natural-Language Processing," arXiv:2212.00237 (2022). https://doi.org/10.48550/arXiv.2212.00237

Spinde, T.; Rudnitckaia, L.; Sinha, K.; Hamborg, F.; Gipp, B.; Donnay, K. "MBIC–A media bias annotation dataset including annotator characteristics." In Proceedings of iConference 2021 (2021). https://doi.org/10.48550/arXiv.2105.11910