Sophia Walton

CS 4501

**Utilizing Post-Hoc Explanations to Identify Bias Words in News Articles**

**Introduction**

In today’s world, media bias is a huge issue that affects our day to day lives. It became especially prevalent during the Trump presidency, when media disinformation reached an all-time high. With events like the Capitol Riot, directly caused by media disinformation claiming that the election results were fraudulent, media disinformation has become a severe danger to society. In order to combat this, some news outlets have tried to work to spread the truth when reporting news. But in this world where humans can’t tell fact from truth, it is important to recognize the biases that exist in our news. The problem with this? Currently, the only ones there to tell people about media bias are news outlets themselves, the very people that they have spent their time reminding us to question. Because of this, it has become increasingly necessary to provide people outside evidence explaining the biases that exist in news. Providing outside sources is a necessary alternative to teach the general public about media bias and how to recognize it.

Machine learning is the solution for helping the public recognize the media bias that exists. By utilizing Natural Language Processing, we have been able to classify different news articles as biased or unbiased in a way that is consistent with the opinions of scholars on the subject[[1]](#footnote-1). However, this doesn’t eliminate the other issue at large - even though we have a model telling us that news is biased, we still lack the explanations necessary to convince the general public that news bias exists.

Using interpretable machine learning, we can take a step further into resolving the public’s understanding of bias. We can not only recognize the existence of bias, but explain where this bias is showing through in an article. This important application of IML could allow us to make real strides towards solving an issue that has had real consequences in the United States and around the world.

**Proposed Dataset**

For this project, I plan to utilize two main datasets. The first dataset that I will be utilizing is the MBIC Dataset that was proposed by Spinde et al. in 2021[[2]](#footnote-2). This dataset was created with the intention of being utilized for IML tasks, including approximately 1200 news articles that were classified and annotated into 2 classes: “biased” “Non-Biased” by human evaluators at Amazon Mechanical Turk. These evaluators also identified the words that led them to this classification as an additional annotation, which can be used as cross validation for the performance of the evaluation methods.

The second dataset is the Media Bias Dataset created by the University of Michigan[[3]](#footnote-3). This dataset was created based on another common dataset of 115000 news sources that has commonly been used in multiclass classifications of topic analysis. The dataset was then modified to include machine and human decided biases of “left”, “moderately left” “neutral”, “moderately right”, and “right”. In order to be consistent with the labels between the two datasets, I will eliminate the political leaning from each of these classifications and reclassify them to the same labels as the first dataset, as seen in the table below.

|  |  |
| --- | --- |
| **Media Bias Dataset Label** | **Associated Label from MBIC dataset** |
| “Left”, “Right” | “Biased” |
| “Neutral” | “Neutral” |

Each news article in the dataset has been left as a URL to a webpage. I parsed the articles at each URL into a string text using a function that utilizes Beautiful Soup to extract only the article data from each URL. I have begun the parsing process on these websites, and due to the different formatting of HTML by site, may end up having to utilize a subset of the dataset depending on the time necessary to correctly format the text from the remaining sites. However, 115,000 examples was a misprint about the dataset, and there were actually 21,500 examples, which after data cleaning led to there being ~16,000 examples in the mediabias dataset. These were then split up into train, test, and validation sets with 13000, 1500, and 1500 examples respectively. From the training data, the vocabulary size was 57,511.

**Proposed Models and Methods**

I performed the initial training and validation on two models: a Random Forest Classifier and a Simple CNN. The Random Forest model was tokenized and preprocessed using the TfidfVectorizer in Sklearn, and the model was performed with 100 trees. The CNN was tokenized using the spacy English tokenizer, and was fine-tuned with 10 epochs each with batch size 16. Ultimately, the performance of the CNN was worse than the Random Forest model and the execution time was much longer, so only the Random Forest model was analyzed when providing explanations. With the success of these models on other binary classification tasks such as sentiment analysis, they make sense as a good starting point to measure the effectiveness of models on this task.

In order to provide the explanations behind the bias these models recognize, I utilized several perturbation-based models to determine the effects that each feature has on the model’s bias prediction (the specifics of the model used to predict the label will be discussed in the next section). From these models, we can compare the results of the experiments both to determine the best method for these explanations and as a sanity check for the accuracy of the model. Finding feature contribution scores from these methods is important for seeing if the models can successfully identify bias words in text.

The first perturbation method that I be implemented is LIME. This model is a good baseline for the interpretability of the model, as good performance on LIME can indicate that news bias is a fairly simple, linear task and would allow us to identify the features that contribute to a biased text more easily. This was implemented using the LIME text python module.

I also evaluated the model using the SHAP values. Due to the long length of news articles in the dataset, calculating the SHAP values is infeasible; the time and memory needed to perform these on such a large dataset would be very high. Instead, I utilized the Sampling Shapley method with 50 random samples of the text in the calculation of feature contribution scores. This model was important to include due to its increased faithfulness to the model when compared to the LIME model. The additional complexity allows for us to further evaluate the performance of the model.

**Experiments and Evaluation**

Due to the lack of prior works checking the best ML method for determining news bias, I will train, validate, and test two different models on the Media Bias Dataset. Additionally, I will test the models on the Out of Domain MBIC dataset. In order to do this, I used both a Convolutional Neural Network and a Random Forest Model with the specifications seen earlier. I found that the Random Forest model performed better than the Neural Network model, as we can see with the accuracy scores in the table below:

|  |  |  |
| --- | --- | --- |
| Model | Random Forest | CNN |
| Validation Accuracy | 0.6781 | 0.5072 |

I then calculated the performance of the Random Forest model on both the in domain and out of domain datasets, finding that it had a test accuracy of 0.6059 on the in-domain dataset and a test accuracy of 0.586 on the out-of-domain dataset.

I then performed the post-hoc analysis of these models using LIME and Sampling Shapley on the out of domain dataset, as this dataset was annotated with the features that contributed to the model explanation.

From these explanations, I will be able to validate the performance of the model against the human-annotated explanations that are provided in the MBIC dataset. From this, I will calculate the precision and recall to determine the cosine similarity of the post-hoc methods with these human-annotated explanations to understand the quality of these annotations.

**Results**

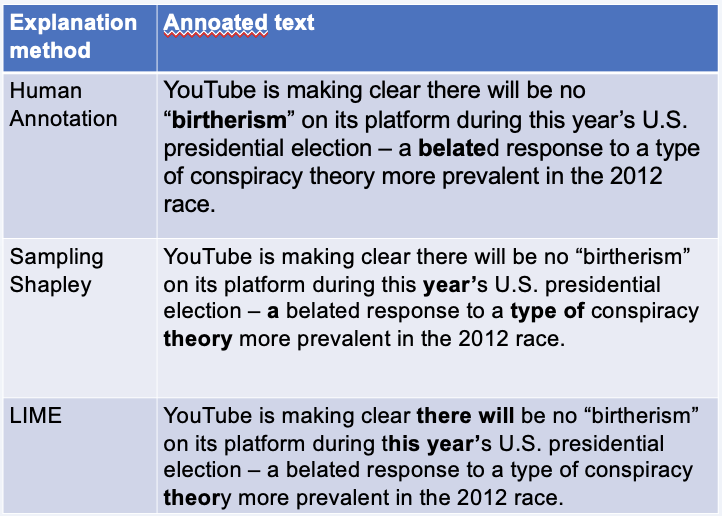
Upon evaluation, the model predictions that led to the decision to utilized the Random Forest model seem to be entirely random guesses. The unbiased label was vastly overrepresented in the predictions, with only 10 percent of predictions being for the biased label.

When evaluating on the LIME label, I found that the most common words used in explanations on the biased class were stop words like “of”, “and” and “the”. Similarly, when evaluating on the Shapley labels, the same kinds of stop words showed up in the annotations. As we can see from the human annotations, the words that correctly identify bias should include contextual words like “dangerous” instead of simply containing these common tokens.

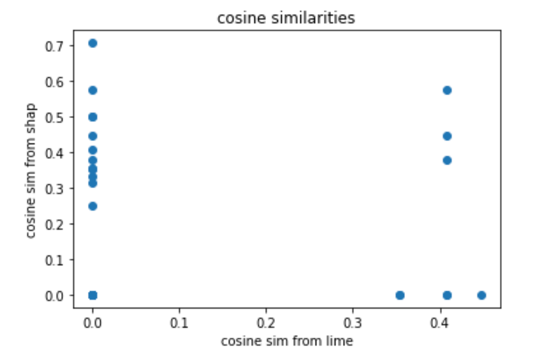
|  |  |
| --- | --- |
| Model | Most common words |
| Human Annotation | “white”, “of”, “radical”, “dangerous” |
| LIME | “of”, “and” “has” “to” |
| Shapley | “the”, “to”, “of”, “a” |

To evaluate the overall performance of the explanation methods in identifying these bias words, I calculated the cosine similarity scores between the human annotated labels and the top 5 tokens in the explanations used.

For the example seen below we found that the model did make the correct prediction of biased. However, in its explanations, we do not see any feature words similar to those found in the human annotation. The LIME model provided the top 5 tokens as “will”, “theory”, “this”, “there”, and “year”[[4]](#footnote-4). Sampling Shapley provided the top 5 tokens to the prediction as “theory”, “year’s”, “a”, “type”, and “of”[[5]](#footnote-5). By contrast, the human annotators identified “birtherism” and “belated” as the context words for their prediction. For this example, we found a cosine similarity score of 0 between the human annotations and both LIME and SHAP. This is fairly representative of the general trends that exist within the dataset.

****

Throughout all the testing examples, there was an average cosine similarity score of 0.00202 for LIME predictions and 0.00435 for Sampling Shapley predictions. Both of these reflect that for the vast majority of examples, there was no overlap between the human annotations and the model feature-importance scores. In the graph seen below, this is seen by the area circled in red, where a large overlap of data exists. From this we can see that there exists a lot more variability in the Sampling Shapley predictions than the LIME predictions, and there are only 6 cases out of 1500 where both LIME and Sampling Shapley perform reasonably well in identifying these words.





Between LIME and Sampling Shapley, there was an average cosine similarity of 0.3755. This represents the fact that they generally contained the same kinds of tokens in their explanations, both selecting a subset of stop words when making their overall predictions.

**Conclusion**

Based on these results, it seems as though models cannot predict whether or not media is biased based on the text. The model really only made random guesses. While the similarity between the human annotated explanations and the machine explanations was low, it is hard to determine whether this is due to the lack of effectiveness of the models or due to the limitations of the explanation methods. The dataset was also challenging – many of the news articles contained specific terms like locations and people that weren’t relevant to other articles, meaning there was a large number of unknown tokens making it more difficult to classify. In order to really draw a conclusion about the methods and their effectiveness, a further study with less limitations would be necessary.

Additionally, while Sampling Shapley performs slightly better than LIME in this case, the performance difference between the two is negligible. The use of random guesses by the model could contribute to the differences we see here. Additionally, these random guesses could account for the larger variability in Sampling Shapley, as Sampling Shapley required many more samples of the model’s prediction for its calculation.

This project was limited by the scope of the dataset. With only 18000 total examples, the ability to effectively train and test the models was limited. In order to measure this effectively, a new dataset with additional examples needs to be created. Additionally, the model architecture could be improved to enhance the performance. The addition of a more complicated language model for embedding would help to resolve the issues that arose from the challenging dataset. A neural network with more convolutional layers would also help to mitigate this issue.

Overall, these results really do not say enough to determine whether bias words can be identified with this method. The limitations seen above are too important to draw any significant conclusions on the topic. From this, the only conclusion we can really make is that measuring political bias is a difficult task for neural network models and humans alike. The ideas here could be measured on other types of data bias that are less limited by the number of examples to see if the methods used here could be a valuable tool in detecting bias.

1. See https://arxiv.org/pdf/1904.01531.pdf [↑](#footnote-ref-1)
2. T. Spinde, L. Rudnitckaia, K. Sinha, F. Hamborg, B. Gipp, K. Donnay “MBIC – A Media Bias Annotation Dataset Including Annotator Characteristics”. In: Proceedings of the iConference 2021. [↑](#footnote-ref-2)
3. Ceren Budak, Sharad Goel, Justin M. Rao, “Fair and Balanced? Quantifying Media Bias through Crowdsourced Content Analysis”, Public Opinion Quarterly, Volume 80, Issue S1, 2016, Pages 250–271, <https://doi.org/10.1093/poq/nfw007> [↑](#footnote-ref-3)
4. LIME feature importance scores (‘will', -0.029252) ('theory', -0.02236), ('this', 0.019847), ('year', 0.015530),), ('there', 0.0111) [↑](#footnote-ref-4)
5. Sampling Shapley feature importance scores: ('theory', 0.054295), ('year’s', 0.008249), ('a', 0.0), ('type', -0.002608), ('of', -0.01205) [↑](#footnote-ref-5)