Detecting Political Bias in News Media

Vivek Gupta
Department of Computer Science
Purdue University
West Lafayette, IN 47907
gupta690@purdue.edu

Srinivasa Arun Yeragudipati
Department of Computer Science
Purdue University
West Lafayette, IN 47907
syeragud@purdue.edu

1 Introduction

News media is an extremely important component of daily discourse and is one of the main sources for the public to form their opinions. However, news articles are written by humans, and inevitably, biases are introduced into them. This can have far-reaching consequences in terms of public opinion and can sometimes be detrimental to law and order, which is evident in today's discussions. Events like the 2016 and 2020 U.S. Presidential elections had widely polarizing statements being made by political leaders, the most exemplary one being Donald J. Trump. Conservative news sources were more favorable towards Trump, and liberal news sources were more antagonistic towards Trump.

The research questions we try to answer in our project are:

- **RQ1:** How feature selection impacts political bias detection? What kinds of embeddings help the models to learn better?
- **RQ2:** Is it possible to eliminate the political bias from the news articles in a systematic way?

To this extent, we formulated our problem statement as *detecting political bias in news media and eliminating it in a systematic way*. Our key contributions can be summarized as

- Sentence-level contextualized embeddings are a better feature representation in detecting political bias than word or paragraph embeddings on news corpus datasets.
- Bias elimination in news articles is possible to an extent, however, this remains an open-ended question.

We have made the code available on GitHub[1].

[1] https://github.com/guptav96/political-bias-detection

2 Related Work

(Fan et al., 2019) focused on the problem of bias detection by creating a new dataset (BASIL dataset) and trying to detect two forms of bias, namely lexical and informational bias. Also, (Baly et al., 2020) focused on the problem of detecting political bias among news articles covering different political topics, focusing on the challenging task of preventing the model from learning the source to infer the bias. (Wei, 2020) focused on learning the news source itself by collecting and training models on a large number of sentences about only one particular topic. Other work includes (Gentzkow and Shapiro, 2010) which tries to predict the leanings of a candidate based on newspaper reports and (Iyyer et al., 2014) which uses recursive neural networks to predict bias on two sentence-level datasets.

(Recasens et al., 2013) developed a model to detect the bias-inducing word from a dataset consisting of Wikipedia edits. Their analysis uncovers two classes of bias: framing bias, such as praising or perspective-specific words, and epistemological bias, i.e. whether propositions that are presupposed or entailed in the text are uncontroversially accepted as true. For this project, we focus only on lexical bias, due to its easy manifestation and detection. (van den Berg and Markert, 2020) focuses on the task of detecting informational bias by defining five baseline models and comparing them with a BERT-based classifier. Previous work on informational bias detection until then has not explored the role of context beyond the sentence.

3 Methodology

The main idea of the project was to get some insight into how good the NLP models are at detecting political bias in news articles. The extended idea was to see if we could use some existing ideas to eliminate the bias, if present. The first step in this direction was to collect relevant data for training.

The task of annotating the data or using mechanical Turk seemed infeasible for the scope of our project. We found two interesting and big datasets suited for our project which we explain in later sections.

The next part of the project was to train a neural network model for bias detection which would output the political stance (Left, Right, Center) of a news article. As discussed in the lectures, representation and structure are the two main aspects of any NLP model. We wanted to focus on the representation aspect of it. Particularly, how different feature representations discussed in later sections affect the performance of our models. One of the key challenges in using these different representations was the computational cost associated with it. Since we trained on two different and huge datasets, the amount of time and resources was significantly higher than we expected. The reasoning behind using these representations was to see how well the bias detection model behaves in the presence of contextualized sentence or paragraph embeddings behave as opposed to word embeddings which do not have any context.

We also tried to answer the open-ended research question of detecting and correcting the bias at the end. We considered this task as a three-step process of: (1) Finding the biased span (2) Identifying the word that introduces the bias (3) Rewording to eliminate the bias. To address this challenge, we considered the works of Jurafsky et al. The BASIL (**B**ias **A**nnotation **S**pans on the **I**nformational **L**evel) dataset seemed like an interesting direction to start on. We faced a lot of challenges while predicting the lexical annotations on our dataset, considering that the paper majorly focused on information bias annotations. The baseline methods discussed in the paper also looked like a compelling direction, and we chose to work on it.

4 Experiments

There were two kinds of experiments we focused on in our project:

- **Exp1**: Detecting the news source and political bias on media articles present in the datasets discussed below in 4.4.
- **Exp2**: Analysing the bias and devising a system to eliminate the bias from the text.

These experiments helped us to answer the research questions we mentioned in the introduction. Exp1 answered our research question on how the feature selection impacts political bias detection, and how different embeddings play a role in inference. Exp2 helped us understand how the bias cues identified by the network can be used to remove the bias from the system.

4.1 Detecting the bias

The main purpose of our first experiment was to predict the type of bias prevalent in different newspaper articles. We performed experiments to predict both the news source and political bias associated with the article. The experiments were performed on both Newspaper Bias Dataset and Article Bias dataset. The other part of the the experiment was to predict the political bias (Left, Center, Right) for each news article. The same model was used for both the cases with different number of output neurons based on the number of classes.

4.2 Eliminating the bias

This experiment focuses on trying to eliminate the bias present in the news article. The key strategies incorporated are: 1. finding the biased word 2. rewording to remove the bias. To find the biased words, we used the Wikipedia baseline, which selects as biased the words which appear in Wikipedia's list of words to avoid which include strong biased words like *racist*, *extremist*, *terrorist*, *neo-Nazis*, *legendary* etc. (Wikipedia, 2023). The biased words are either removed from the sentence or reworded using one of wordnet (Fellbaum, 1998) synsets that shifts the output to other direction or neutral side (based on what is required).

4.3 Method Variants

We were interested in comparing the performance of our bias detection system in three different settings. These settings were mainly defined by the choice of the underlying feature representation of the input text. To put in simple words, we tried three different contextualized embeddings, namely word-level embeddings, sentence-level embeddings and paragraph-level embeddings. These are briefly discussed below:

Word Embeddings: Word embeddings are primarily used to capture the semantic relationships between words. It can be thought of as points in a high-dimensional vector space obtained using a shallow neural network, where the vectors close together have similar meanings based on context,

and word-vectors distant to each other have differing meanings. We used word2vec (Mikolov et al., 2013) embeddings for our experiments with their TF-IDF where each weight gives the importance of the word with respect to the training corpus, and decreases the influence of the most common words. We consider these embeddings as a strong baseline or feature for our tasks.

Sentence Embeddings: The idea of word embeddings can be extended to sentences where instead of learning feature representations for words, we learn it for sentences. The sentence embedding model takes into account not only the context in which each word appears, but also the context of the entire sentence. We used commonly used sent2vec (Pagliardini et al., 2017) embeddings to construct the feature representation of our input text.

Paragraph Embeddings: Doc2Vec (Le and Mikolov, 2014) algorithm introduced in 2014 outperformed the simple-averaging of word2vec vectors. There are two different implementations available for doc2vec. For simplicity, we used only one implementation namely, Paragraph Vector - Distributed Bag of Words (PV-DBOW) for all our experiments.

4.4 Datasets

We used two publicly available media bias datasets for training. These datasets differ in the news sources and the length of annotated content. The datasets are explained below.

4.4.1 NewB Dataset

The Newspaper Bias dataset (Wei, 2020) is a collection of over 200,000+ sentences regarding Donald Trump from eleven news sources. This dataset covers the political views of eleven popular media sources, capturing more nuanced political viewpoints than a traditional binary classification system does. This dataset is different from other dataset in the fact that the labels are a collection of 11 newspapers across the political spectrum. Each of the news sources in the dataset is assumed to have a specific bias. For example, the New York Times has a liberal leaning and the New York Post is assumed to towards the conservative spectrum. Also, we have 24,000 sentences for each newspaper, which amounts to a large amount of data.

4.4.2 Article Bias Dataset

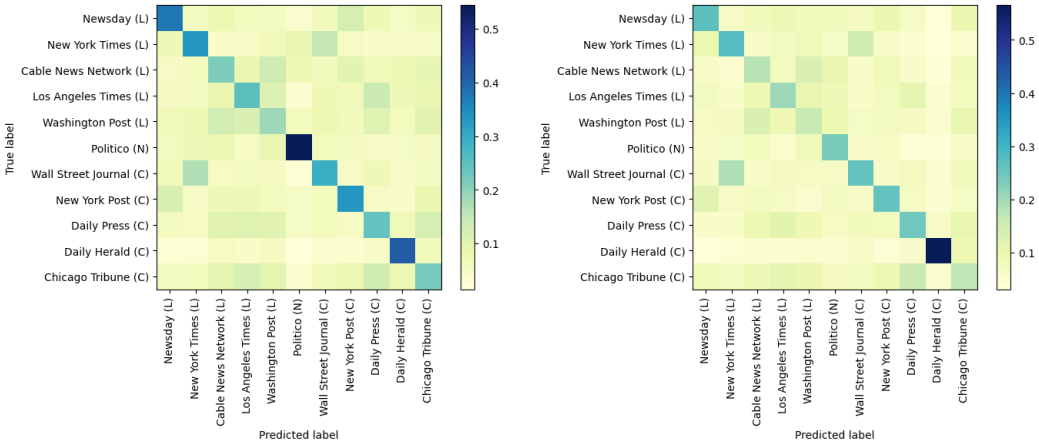
Article Bias Dataset (Baly et al., 2020) consists of 37,554 news articles about various topics from various newspapers. The documents are collected from 73 news sources and contains 109 topics. Each of the documents is annotated as either left, centre or right. This dataset stands out as it provides annotations of political ideology for individual articles, which ensures high-quality data for both training and testing, which is in contrast with distant supervision approaches used in most previous research. Also, in this dataset we observe that some articles have biases that are not consistent with the sources. The motivation behind using this dataset is to capture diversity and see if the models we train could perform well on a slightly similar, yet a different dataset.

Dataset	Task	N	c
Article Bias	Political Bias	37,554	3
NewB	News Source	264,000	11

Table 1: Comparison of the two datasets. N is the dataset size and c is the number of classes.

4.5 Training Details

For this project, we primarily used the PyTorch Framework. We used Google Colaboratory to conduct all our experiments. The embedding dimension for each article was fixed at 384 for sentence and paragraph embeddings. We used gensim module to train our own paragraph embeddings. For word embeddings, we used embedding dimension of 300. We trained all the models using a feed-forward neural network with 3 hidden layers of size 128. The loss function used for optimization was categorical cross-entropy loss. We used the Adam (Kingma and Ba, 2014) optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and $\epsilon = 1e^{-8}$. We used a learning rate of $1e^{-4}$. All these settings were kept consistent throughout all the runs of our experiment to prioritize on comparing the performance difference between different feature representations as we discussed in 4.3. We also apply dropout (Srivastava et al., 2014) to the output of input and hidden layers to prevent model overfitting. For the experiments, we use a rate of $P_{drop} = 0.2$. We trained each model for a minimum of 30 epochs.



(a) Sentence embeddings

(b) Paragraph embeddings

Figure 1: Confusion matrix of predicted and ground truth news sources trained on NewB

4.6 Results

The results of Experiment 1 are discussed below. For detecting the source of news article on NewB dataset, we calculate top-n accuracies per class. Tables 2 shows the top-1,2,3,4,5 accuracies for detecting 11 news source on NewB dataset with different embeddings.

Features	Top-n Accuracy (%)				
	n=1	n=2	n=3	n=4	n=5
word2vec	26.2	40.7	51.3	60.1	68.4
sent2vec	**31.9**	**47.5**	**58.1**	**67.4**	**74.3**
doc2vec	25.5	41.1	52.2	61.6	69.6

Table 2: Top-n accuracies for each embedding on NewB dataset. Note that the dataset has 11 classes.

We also predicted the political sentiment of a news article (Left, Center, Right) on both the datasets whose results are shown in the Table 4. Furthermore, we also display the confusion matrices of predicted labels in the form of a heatmap in Figure 1 using sentence and paragraph embeddings as feature representations.

We notice that sentence embeddings are rich feature representations in detecting political bias for the datasets we considered. The main reason behind this is that they are able to capture the high-level context of a sentence. Moreover, the performance is better on NewB dataset since the articles contained in the dataset are typically 1-2 sentences long.

From the confusion matrix, we can observe that Politico, New York Times and Daily Herald are the easiest to classify. The argument behind this could be that Politico uses a larger variety of words than other classes as shown in its larger vocabulary size, and New York Times has the shortest average sentence length. There is also a high false positive rate for Wall Street Journal on New York Times sentences, likely because both newspapers tend to have short sentences with an average of only 18 words per sentence.

Model	Accuracy (%)		
	word2vec	sent2vec	doc2vec
NewB	56.2	**62.5**	58.7
Article Bias	67.8	**79.4**	70.4

Table 3: Accuracies for each embedding on NewB and Article Bias datasets with 3 classes (Left, Right and Centre).

We now discuss the results from experiment 2 briefly with a liberal example from NewB dataset shown in Table 4. Finding the biased word using Wikipedia's baseline gives us *extremist*. Trying a few transformations of sentences reveals that it is actually possible to remove/shift the bias.

5 Discussion

The project helped us gain valuable insights on different areas concerning the representation of language and structure modeling. It also helped us learn how different types of biases exist in the world. We also aimed to detect one such kind of bias using machine learning models we trained from scratch. Based on our experiences, we found that it is possible for us to detect biases in news articles. Also, we found out that to a significant

Sentence	O/T	L	C	N
he called trump a puppet a novice and an extremist	O	**0.53**	0.07	0.39
he called trump a puppet and a novice	T	**0.48**	0.08	0.44
he called trump a puppet	T	**0.52**	0.05	0.44
he called trump a genius	T	0.41	0.04	**0.55**

Table 4: Results from Experiment 2 with O representing the original sentence and T represents the transformed sentences. L here refers to Liberal, N refers to Neutral and C refers to Conservative

extent, we can try to remove the biases as well.

Some of the main things that worked out for us is the availability of the datasets, and the knowledge of language representations from the lectures. If we had more time, say 6 months, we would actually try to create our own dataset by scraping articles from mostly the same news sources but cover different topics. In a sense, we would like a dataset which has the best of both the NewB and the Article Bias datasets, i.e. have more topics and more articles for each topic. One more thing we would have loved in our project, if time permitted, was to do more analysis on experiment 2 and build an end-to-end model that could perform both the experiments. We hope that we would be able to achieve this in the future.

References

Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We can detect your bias: Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '20, pages 4982–4991.

Lisa Fan, Marshall White, Eva Sharma, Ruisi Su, Prafulla Kumar Choubey, Ruihong Huang, and Lu Wang. 2019. In plain sight: Media bias through the lens of factual reporting. *arXiv preprint arXiv:1909.02670*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Matthew Gentzkow and Jesse M Shapiro. 2010. What drives media slant? evidence from us daily newspapers. *Econometrica*, 78(1):35–71.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Esther van den Berg and Katja Markert. 2020. Context in informational bias detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6315–6326.

Jerry Wei. 2020. Newb: 200,000+ sentences for political bias detection. *arXiv preprint arXiv:2006.03051*.

Wikipedia. 2023. Wikipedia - manual of style/words to watch. *Wikipedia*.