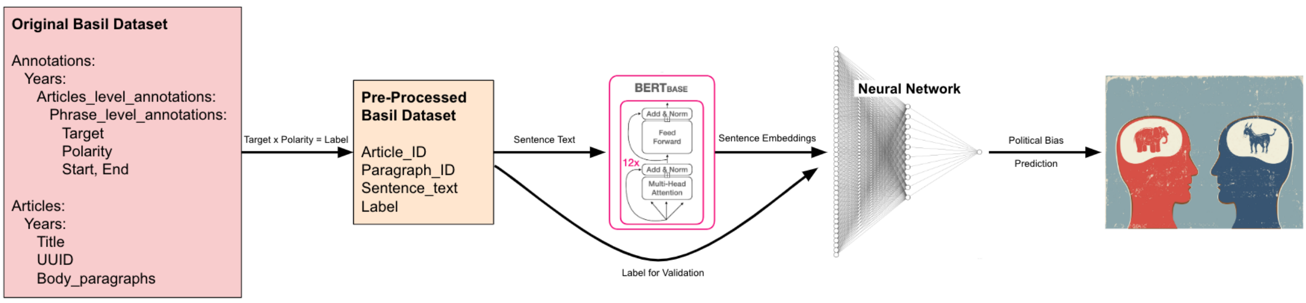
Political Media Bias Prediction for Resource-Limited Devices

John Cashy [*] **Sofia Beachman** [*] **Ian Fischer** [*]

[*] *University of Michigan, Ann Arbor, MI 48105, USA (e-mail: {jtcashy, sofiabe, ifisch} @umich.edu)*

Abstract: This paper discusses the need for bias detection in news media, specifically for low-powered devices such as phones and tablets. The study utilizes natural language processing techniques to predict the political bias present in news articles at the sentence level. Previous studies using the BERT transformer, bag-of-words, and SimCSE learning framework have shown promising results in determining sentence-level bias. This paper discusses recent studies using the RoBERTa language model, the LSTM neural network, and the Multilayer Perceptron models (MLP). The BASIL dataset, consisting of 300 annotated articles from three political news sources, was used for sentence-level bias prediction. To process the data, the bias score for each annotation was calculated using the political leaning of the main entity and the polarity of the text. The paper concludes that natural language processing techniques have the potential to detect political bias in news media and provide users with knowledge of the biases present in their media.



1. INTRODUCTION

In the United States, the political landscape is split into two dominant ideologies - liberal and conservative - represented by the Democratic and Republican parties respectively. These two groups promote distinct policies that mirror opposing views on various social and political matters. Mirroring the political polarization growing in the United States in recent years, there has an increase in political bias present in media news. The unique position of media news to impact public perception through how it both presents factual information and the tone of its writing illustrates the need for bias detection in the news. However, typical state-of-arc detection solutions require significant computational resources, which lower powered devices like phones and tablets can not support, thus there is additionally a growing need for solutions built for these resource limited devices.

The goal of this project is to provide resource limited device users with knowledge of the political biases present within their media. We aim to accomplish this by utilizing natural language processing techniques to analyze news articles to generate a prediction of political bias at the sentence level.

2. RELATED WORKS

With the increasing prevalence of bias in news media and the continual growth of natural language processing technology, the advancement of article and sentence-level bias detectors has progressed in recent years.

2.1 Informational and Lexical Bias

The BERT transformer firstly demonstrated success in determining sentence level bias, used by following studies as a baseline. In the paper, "In Plain Sight: Media Bias through the Lens of Factual Reporting," researchers looked at the difference between lexical and informational bias [1]. While most of the political bias detection models from previous studies focused on lexical bias, looking at word choice and syntactical factors, they chose to ascertain the impact of informational bias. Informational bias focuses on spans of biased phrases or sentences that convey information about the main entity of an article that may sway the opinion of the reader. In this study, researchers composed the BASIL dataset, consisting of articles from three politically varying news sources, in order to determine the use of informational bias. The study concluded, in accordance with the selected articles, that news sources are more likely to incorporate informational bias, rather than lexical bias. Additionally, the study found that informational bias tends to be evenly distributed throughout an article, while lexical bias typically occurs only in the beginning. Lastly, the researchers fine-tuned a cased BERT model for predicting political bias in news articles.

		NYT	FOX	HPO	All
# Articles		100	100	100	300
# Sentences		3,049	2,639	2,296	7,984
# Words		91,818	70,024	62,321	224,163
# Annotations		636	573	518	1,727
Sentences / Article		30.5 ± 13.8	26.4 ± 10.2	23.0 ± 11.0	26.6 ± 12.2
Words / Sentence		30.1 ± 14.0	26.5 ± 12.4	27.1 ± 12.5	28.1 ± 13.2
Annotations / Article		6.4 ± 4.1	5.7 ± 3.8	5.2 ± 3.5	5.8 ± 3.8
Bias Type	*Informational*	468 (73.6%)	421 (73.5%)	360 (69.5%)	1,249 (72.3%)
	Lexical	168 (26.4%)	152 (26.5%)	158 (30.5%)	478 (27.7%)
Aim	*Direct*	574 (90.2%)	485 (84.6%)	462 (89.2%)	1,521 (88.1%)
	Indirect	62 (9.8%)	88 (15.4%)	56 (10.8%)	206 (11.9%)
Polarity	*Positive*	112 (17.6%)	89 (15.5%)	110 (21.2%)	311 (18.0%)
	Negative	524 (82.4%)	484 (84.5%)	408 (78.8%)	1,416 (82.0%)
Annotations in quotes		205 (32.2%)	299 (52.2%)	217 (41.9%)	721 (41.8%)

Fig. 1. BASIL Statistics

2.2 BOW and Contrastive Framework Models

Additionally, in the study "Detecting Bias in News Articles using NLP," researchers found that the SimCSE learning framework proved to be more effective in detecting sentence-level bias than a TensorFlow deep neural network using a TF-IDF weighting factor [2]. In this study, researchers created a baseline TensorFlow deep neural network using bag-of-words to extract features from input news articles. To improve the baseline model they incorporated TF-IDF (term frequency-inverse document frequency) as a weighing factor. The TF-IDF weighting factor creates a more realistic model by highlighting not just the frequency of a word, but also the uniqueness of that word to a specific news source. Additionally, the study implemented a K-Means clustering algorithm using TF-IDF that illustrated similarities between sentences of various news sources. Lastly, the researchers built a SimCSE model, a contrastive framework that uses positive and negative data samples to correlate similar sentences. A logistic regression classifier and Adam optimizer were used to correctly identify these samples. The study found the baseline TF-IDF and K-Means models to be relatively ineffective at detecting sentence-level bias with low accuracies. They attributed this to the lack of diverse and unique words present throughout the articles, making it difficult for the models to classify.

2.3 Recent Studies

The use of the RoBERTa language model in tandem with a Bi-LSTM layer and PDTB discourse relation layers has produced successful results in determining bias from combining both the article context and sentence semantics [4]. Similarly, the LSTM neural network proved useful in creating a 2-dimensional ideological-bias and content quality measurement for Tweets [5]. Further, Multilayer Perceptron models (MLP) achieved higher accuracy in bias detection than Recurrent Neural Networks (RNN) on certain datasets, but failed to yield these results on live data [3].

3. DATA PREPROCESSING

3.1 The BASIL Dataset

To predict sentence-level political bias, we are working with the 2nd generation of the BASIL dataset. The dataset contains 300 annotated articles from three political news sources, New York Times (liberal), Fox News (conservative), and Huffington Post (neutral). The article files contain information about the main entities, the main events, the news source, and the text of the article broken down into body paragraphs. In addition, BASIL contains correlated annotation files that describe the political bias of specific spans of text within paragraphs of a given article. The dataset includes annotations for both informational and lexical bias. Therefore, we consider both types of biases in our predictions. These annotations include a variety of defining characteristics, most importantly, the polarity of the bias (negative or positive), the associated paragraph id, the target entity of the sentence, and the given text.

3.2 Condensing the Data

In order to utilize the BASIL dataset for our task, we combined attributes from each biased annotation to calculate an overall bias score. We collected the main entity of the text and the polarity for each annotation. To calculate the political leaning of the sentence we compared the political label of the main entity (liberal, conservative, or neutral) with the polarization (positive or negative). For example, a sentence with the main entity Marco Rubio (labeled as conservative), combined with the polarity of "negative," will denote a liberal leaning bias. We chose to label "conservative", "liberal", and "neutral", as -1, 1, and 0, respectively. The BASIL dataset does not provide the associated political leaning of each main entity. Therefore, we collected all main entities present within the dataset and hand-labeled them as either "conservative", "liberal", or neutral, using -1, 1, or 0, as mentioned above. Annotations without any main entities or without any polarity score are not included the final dataset.

3.3 Dataset Results

To prepare the dataset for binary classification, all sentence bias scores greater than 0 (liberal leaning) are re-valued to 1. Similarly, all sentence bias scores less than 0 (conservative leaning) are re-valued to -1. Lastly, all neutral sentence with bias scores of 0 are dropped from the dataset. After preprocessing, the new dataset contains 1723 unique sentences pulled from 297 separate articles, each with bias labels -1 or 1. There were a total of 795 bias scores of value -1, therefore %46 of sentences were classified as conservative. The other 928 sentences held a bias score of 1 and were classified as liberal. We utilized SKLearn's train_test_split to divide the data into training, validation, and test data.

4. METHODOLOGY

In order to detect sentence-level political bias, we are constructing two transformer based models. The baseline

article_id	paragraph_id	paragraph_txt	final_label
c356e514-a20d-45e3-b1ad-159636013822	p0	Mary Cheney, the younger sister of Liz Cheney, a Wyoming Senate candidate, sharply criticized her sister‚Äôs stance on same-sex marriage and urged her own Facebook friends to share the message.	1
c356e514-a20d-45e3-b1ad-159636013822	p1	Posting on Facebook on Friday evening, Mary Cheney, who is gay and married her longtime partner last year, wrote: ‚ÄúFor the record, I love my sister, but she is dead wrong on the issue of marriage.‚Äù	0
c356e514-a20d-45e3-b1ad-159636013822	p2	Their father, former Vice President Dick Cheney, supports same-sex marriage, and the younger Cheney echoed some of his language on the issue when she added, ‚ÄúFreedom means freedom for everyone.‚Äù	0
c356e514-a20d-45e3-b1ad-159636013822	p3	‚ÄúThat means that all families ‚Äî regardless of how they look or how they are made ‚Äî all families are entitled to the same rights, privileges and protections as every other,‚Äù Mary Cheney wrote.	0
c356e514-a20d-45e3-b1ad-159636013822	p4	Earlier Friday, Liz Cheney revealed her position on same-sex marriage, a topic she has kept relatively quiet about since declaring her candidacy in July against incumbent Senator Mike Enzi, Republican of Wyoming.	0
...	...	...	...

Fig. 2. Input Dataset

model will be used to compare the efficacy of our experimental model for resource-limited devices. In this section, we will discuss the main aspects of our approach: model architecture and common training parameters between the two models.

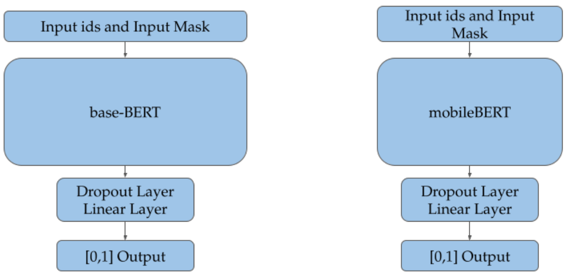


Fig. 3. Model Architecture

4.1 Model Architecture

Baseline BERT The baseline model was constructed using the uncased base BERT model which feeds into a neural network. The BERT transformer model utilizes a bidirectional encoder to train on the text and produce embeddings. We input the tokenized sentences into BERT, and utilize the associated bias score (0 or 1) for the neural network. In doing so, we create a model that outputs the bias of a given sentence. BERT's success as a language model stems from this bidirectional training that looks at every token of the input at one time, and provides a more conceptual understanding of the text compared to directional models. In our initial proposal we planned on utilizing a cased BERT model for named entities, however, we determined that uncased BERT works more appropriately with our input dataset, as many of the input tokens were unknown with Cased BERT. The uncased base BERT model has 12 layers, 768 hidden layers, 12 heads, and 110 million parameters. Due to the size of base BERT, the model takes significant time and computational resources to train. The BERT embeddings were then fed into our MediaBiasDataset model, a neural network with one dropout and one linear layer.

Experimental MobileBERT The experimental model will mirror the architecture of the baseline but will utilize the uncased MobileBERT model instead. MobileBERT is a transformer model similar to BERT that incorporates an inverted-bottleneck technique in order to decrease resource requirements [6]. The MobileBERT model only has 25 million parameters, a 77% decrease from the total parameters of the baseline BERT model. The significant decrease in size will allow the model to train and hyper-tune more quickly and with less resources than base BERT. We will utilize the baseline model to accurately compare the information retention between the two models.

4.2 Common Model Information

While the underlining architecture of the baseline and experimental model differs, certain parameters remain the same between the two models. For this project, the loss function utilized to train both models was Binary Cross-Entropy. Additionally, for both models, the batch size utilized in the training, validation, and test sets remained the same at 32 data points per batch.

5. EXPERIMENTS

5.1 Model Training

Dataset Creation In order to train the models, we first had to split our dataset into training, validation, and test sets to accurately train and evaluate the models' results. Using random sampling, we selected 70% of the modified dataset to become our training set. We then separated the remaining portion of the dataset, using 18% to become our validation set and 12% to become our test set. Furthermore, the training, validation, and test sets were stratified based on the Conservative-Liberal label, ensuring a consistent ratio of each label in all three datasets. This means that the training, validation, and test sets all had a ratio of 53.85% Liberal to Conservative labels. Lastly, all the datasets were batched using a size of 32 for ease of tuning.

Hyperparameter Tuning Both models underwent a grid search to tune hyperparamenters. The hyperparameters tested were learning rate and the dropout layer percentage. Due to the differing structures of our two models, each model had different values that were searched. For instance, the reduced size of pretrained mobileBERT requires higher learning rates than base-BERT to be properly tuned. Furthermore, mobileBERT takes longer to converge than base-BERT, thus, the experimental model was trained over 10 epochs compared to the baseline being trained over 5 epochs. The results of hyperparameter testing are included below.

From the hyperparameter testing, we selected the model that achieved the best validation accuracy. Thus, our best baseline model trained over 5 epochs had a learning rate of 5e-05 and a droupout percentage of 10%. Our best experimental model trained over 10 epochs had a learning rate of 5e-04 and a dropout percentage of 12%.

5.2 Model Evaluation

In order to compare the effectiveness of the resource-limited model to the base-BERT model, we considered three metrics to determine the success of our model. Firstly, we considered the accuracy of the model on the testing set to compare how successful the model was

Baseline

Learning Rate	Dropout Percentage	Validation Accuracy
3e-05	.10	0.7096
3e-05	.15	0.7064
3e-05	.20	0.6838
5e-05	**.10**	**0.7129**
5e-05	.15	0.6903
5e-05	.20	0.6870
7e-05	.10	0.5387
7e-05	.15	0.7032
7e-05	.20	0.7064

Experimental

Learning Rate	Dropout Percentage	Validation Accuracy
5e-04	**.12**	**0.6354**
5e-04	.15	0.5935
6e-04	.12	0.5709
6e-04	.15	0.6322
7e-04	.12	0.6129
7e-04	.15	0.5516
8e-04	.12	0.5387
8e-04	.15	0.6161

Fig. 4. Hyperparameter Search Results

at the classification task. However, we also valued the efficiency of the models prediction of the testing set. Thus, we additionally measured the amount of GPU memory that the model consumes on classifying the test set and the amount of time the GPU was in use. We recorded these metrics on the each of the models selected from hyperparameter tuning.

	Baseline	Experimental
Testing Accuracy	**0.6135**	0.6039
Testing GPU time	1126.00 ms	**379.47 ms**
GPU memory usage	895.13 Mb	**763.37 Mb**

Fig. 5. Test Set Results

5.3 Model Results

As shown in figure 5, our experimental model produced comparable testing accuracy to our baseline model with a difference of only .96%. Additionally, our experimental model performed significantly better on GPU time and GPU memory usage, outperforming the baseline model by 746.53 ms and 131.76 Mb, respectively. Thus, it can be determined that we achieved our goal in constructing a lighter model that can successfully classify biased sentences with the same effectiveness as a complex model.

6. FUTURE WORK

While the results of this study are promising, there is still much work to be done in the field of bias detection in news media. Some potential future directions or work include:

6.1 Aggregate sentence-level bias predictions into paragraphs or entire article-level bias predictions

In this study, the prediction of political bias was made at the sentence level. However, it is possible to aggregate these sentences to determine the overall political leaning of each paragraph, or further aggregate to determine the political bias of an entire article. This would provide users with a more comprehensive understanding of the political bias present in the media they consume.

6.2 Gather new data and try to improve our network accuracy

While the BASIL dataset used in this study was useful for training and testing the models, it is limited in size and scope. Gathering more annotated data from a wider variety of news sources could improve the accuracy of the models and make them more robust.

6.3 Develop a Chrome extension or mobile app to perform on-the-fly bias detection for users on social media or the internet

One potential application of this technology is developing a Chrome extension or mobile app that can detect on-the-fly bias for users on social media or the internet. This would allow users to quickly and easily identify the political biases present in the media they consume, and could potentially help to reduce the spread of misinformation and propaganda.

7. CONCLUSIONS

In conclusion, the use of natural language processing techniques to detect political bias in news media shows great promise, and there is still much room for future research in this field. By continuing to improve the accuracy of our models and developing new applications for this technology, we can help to ensure that users have access to unbiased and accurate information.

7.1 Breakdown of Work

John Developed the model architecture, devised hyperparameter grid search, and implemented evaluation metrics.

Sofia Provided background research on various NLP techniques and model architectures. Assisted in data preprocessing and cleaning. Worked on analysis and written report.

Ian Worked on data cleaning and preprocessing, made the slide deck for our presentation, assisted in writing reports and generating figures for results

REFERENCES

[1] L. Fan, M. White, E. Sharma, R. Su, P. K. Choubey, R. Huang, and L. Wang, "In plain sight: Media bias through the lens of factual reporting," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Nov. 2019.

[2] M. U. Nadeem and S. Raza, "Detecting Bias in News Articles using NLP Models," Stanford CS224N Custom Project.

[3] M. Vu, "Political News Bias Detection using Machine Learning."

[4] Y. Lei, R. Huang, L. Wang, and N. Beauchamp, "Sentence-level Media Bias Analysis Informed by Discourse Structures," in Conference on Empirical Methods in Natural Language Processing, 2022.

[5] Z. Chao, D. Molitor, D. Needell, and M. A. Porter, “Inference of Media Bias and Content Quality Using Natural-Language Processing,” Dec. 2022.

[6] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, “Mobilebert: A compact task-agnostic Bert for Resource-limited devices,” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Jul. 2020.