Flagging Political Bias in News Headlines

Cherie Fernandes | cferns@uchicago.edu

Executive Summary

Right or left? Like a less-fun dating app with about the same success-rate, this project attempts to construct a classifier which identifies political bias in newsmedia. I consider a repository of news articles from the last odd decade, pre-classified by experts at AllSides, and built out a transform model (distiliBERT) which identifies left, center, or right lean in a given news headline. The resulting model is of poor general accuracy, struggling to differentiate between center and left leaning text, but does effectively flag pieces with a right lean and outperform a baseline model (TF-IDF + logistic regression). Such a classifier could be useful to editors and newsrooms attempting to flag their own bias, rather than merely consumers, for whom resources that gauge bias focus on biased news *sources*, rather than the text itself.

Problem

Public faith in the media is at an all time low. News reporting—a matter which could, theoretically, be entirely objective—is increasingly shaped by political narratives, and newsrooms allow columnists to crowd in. Accordingly, a model that classifies news articles as left, center, or right-biased on a text, rather than source, level could help readers *and* journalists recognize bias, fostering more critical media consumption and creation. I opt to look at news headlines, rather than the entire text of a given article due to time and computing constraints, but headlines still distill the main features and tone of a given article for a consumer, and thus the natural element to scan for bias. Ideally, such a tool could be used as a sense-check for editors to gauge bias in their own work prior to publication.

Project References

Note that these are largely similar to that in my report, with revised “implications” sections to demonstrate what role each paper actually eventually played in my project.

Project Reference 1: *We Can Detect Your Bias: Predicting the Political Ideology of News Articles,* 2020 [URL]

This paper **classified left/right/center bias in the text of news articles**, drawing on Kulkarni et al., 2018, considering all news topics (*Sports, Media & Entertainment*, *Covid-19*, etc.).

1. **Data:** News articles downloaded from AllSides through 2019, which provides manually annotated articles with political bias labels (left, center, right). Ultimately 34,737 articles from 73 news sources covering 109 different topics.

2. **Data preprocessing:** Removed explicit markers such as author names, media names, and other metadata that could reveal the source. Articles from the same media source were separated in different train/test splits to prevent the model from simply learning a source's ideology.

3. **Models considered and used:** Baseline models included **LSTM** (Long Short-Term Memory), a recurrent neural network that captures word sequences, and **BERT** (Bidirectional Encoder Representations from Transformers), a transformer-based model known for strong text representations. For debiasing, applied **adversarial adaptation** (a media classifier that forces the model to extract bias-relevant information while ignoring media-specific cues), and **triple loss pre-training** such that ideologically similar articles are grouped closer together, regardless of their media source.

4. **Metrics for performance evaluation:** A simple accuracy %, Macro F1 Score (across all classes), and MAE.

5. **Evaluation:** At baseline, BERT performs best with 80.19% accuracy, but poorly under a deliberate media bias split (test set only includes articles from news sources not seen during training, forcing the model to generalize beyond media recognition), suggesting they **memorized media sources rather than learning article-level bias**–this is an important roadblock. Applying external data, such as the twitter bios representative of each news article's readership (BERT with TLP + Twitter bios) sees a 72.00% accuracy in the media-bias split.

6. **Results:** Without debiasing, these models effectively classify article text, but use news sources as a proxy. This suggests that metadata was not sufficiently effaced. Debiasing enhanced model performance.

7. **Implications:** Ultimately, I failed to release that I don't have access to nearly as much compute power as the authors of this study, and was unable to apply BERT (or even distillBERT) models to the lengthy text samples (500-800w) they used. Accordingly, their 70% accuracy for a BERT model on C/L/R is no longer a relevant benchmark. However, I was able to draw on their use of BERT vs a baseline logistic model in my own project, which looks at a much smaller text size. This article also gives me a sense of possible

extensions to my own project via debiasing, which I wasn't able to accomplish and understand in this timeframe.

Project Reference 2: *Detecting Bias in News Articles using NLP Models*, 2022 [URL]

Develops a classifier that can detect the **news source of a given sentence (of a particular topic)**, which can later be used to infer political bias.

8. **Data:** NewB dataset compiled by J. Wei (2020), comprised of 250,000 sentences from 11 news sources labeled by source, as well as News sources categorized by bias (5 liberal, 1 neutral, 5 conservative). All sentences are topical to Donald Trump.
9. **Data preprocessing:** Converted raw text files into CSV using pandas, tokenized sentences using NLTK tokenizer, stored training data as lists of strings (JSON format), and standardized input structure for each model.
10. **Models considered and used:** First Bag-of-Words with DNN, a simple neural network with four layers. Limited because it ignores word order and structure. Improved with TF-IDF (Term Frequency–Inverse Document Frequency) Weighted DNN, which prioritizes important words over frequent but uninformative words. Applied K-Means Clustering, using TF-IDF weighted word embeddings and initially trained with k=11 clusters. Also applied SimCSE model trained using BERT embeddings (loss fn = Cosine similarity + Cross entropy loss), which compares similarity between sentences based on their contextual meaning rather than just word occurrences.
11. **Metrics for performance evaluation:** BOW and TF-IDF evaluated by accuracy %, K-means clustering evaluated by distribution of sentences across clusters, and SimCSE was evaluated with a Spearman correlation coefficient (A test sentence was compared with multiple sentences from each news source, and the one with the highest average correlation was considered the model's prediction)
12. **Evaluation:** BOW and TF-IDF saw low accuracy (15%), and K-means clustering failed to cluster meaningfully, with 95% of the data in the same cluster for both k =11 and the adjusted k = 2. SimCSE ultimately had the highest accuracy at 24.3%, likely best for sources with a consistent sentence structure.
13. **Results:** SimCSE outperformed all other models because it captured sentence structure and framing choices, not just word frequency. However, one sentence of data does not seem enough to make a reliable prediction about news source—short paragraph structure or even article headline may have been more prudent.
14. **Implications:** Ultimately, this was a good template for classification of sentences, rather than large blocks of text; it was more accessible than the previous article and better for

understanding exactly what tokenizing and featurization are as concepts in NLP. The article helped my understanding of BoW and TF-IDF and its observations surrounding its limitations with similar vocabulary/key terms. I decided that SimCSE wouldn't be a good fit for me, given that my classifier is binary or three-way compared to this study's eleven, but I did use the toolkit in DistilBERT.

Description of Solution

I used the QBias ("A Dataset on Media Bias in Search Queries and Query Suggestions") dataset [X], which contains **21,747 news articles scraped from AllSides balanced news headline roundups [X] in 2012-2022**. The AllSides balanced news feature three expert-selected U.S. news articles from sources of different political views (left, right, center), often featuring spin bias, and slant other forms of non-neutral reporting on political news. All articles are tagged with a bias label by expert annotators based on the expressed political partisanship, left, right, or neutral. Collected data further includes headlines, dates, news texts, topic tags (e.g., "Republican party", "coronavirus", "federal jobs"), and the publishing news outlet. Given that my intention is to detect bias in the newsroom producing a piece of content, I limited my classifiers to the text, heading, and bias columns.

As described under "roadblocks", I initially wanted to build the classifier (C/L/R) based on the text of each article (500-800w), but ultimately opted to use solely the headlines. I also initially wanted to consider only science communication, but found that the total number of such articles was relatively low, and that fitting my models to that restricted set of data worsened performance (see below).

I used the following software throughout the project:

- Data collection: selenium, requests / BeautifulSoup
- Data preprocessing: spaCy, scikit-learn
- Feature extraction: TF-IDF, transformers (DistilBERT), datasets, torch
- Model training & evaluation: scikit-learn, tensorflow, torch, SHAP

I began by balancing the dataset and doing an initial featurization of the text and heading columns. I then trained and fit the following models:

- TF-IDF + Logistic Regression (Baseline) - Text
- TF-IDF + Logistic Regression (Baseline) - Headlines
- TF-IDF + Random Forest Classifier - Headlines
- TF-IDF + SVM - Headlines
- Fine-Tuned Transformer Model (DistilBert) - Headlines

To each of the following datasets:

- Full 2012-2022 dataset with C/L/R classification
- Full 2012-2022 dataset with Biased/Unbiased classification
- Only medicine-specific articles (filtered set of “topics” tags to {'Coronavirus', 'Public Health', 'Healthcare'}) with Biased/Unbiased classification

The F1 scores are as follows:

F1 Scores across Classifiers

	Logistic (Text)	Logistic (Header)	RF (Header)	SVM (Header)	Transform (Header)
Full, C/L/R	*0.45*	**0.38**	0.39	0.39	**0.40**
Full, B/U	*0.60*	0.55	0.54	0.55	0.56
Medical, B/U	*0.58*	0.54	0.47	0.51	0.51

Clearly, this didn’t go very well–the header-based classifiers don’t do significantly better than random chance. The distilBERT transform model, which is built out in Final_Project (Tripartite).ipynb, is ultimately strongest relative relative performance, with the following classification report:

DistilliBERT Model on Headers – Classification Report

	precision	recall	f1-score	support
left	0.44	0.23	0.30	850
center	0.47	0.23	0.31	851
right	0.37	0.73	0.49	851
accuracy			0.40	2552
macro avg	0.42	0.40	0.37	2552
weighted avg	0.42	0.40	0.37	2552

See also our baseline model (TF-IDF + logistic regression) below for a benchmark. We might have used Baly et. al. as a benchmark (70% accuracy), but they classified entire texts rather than merely headlines–which, based on the difference in my F1 with the simplest model (0.45 vs 0.38) is a non-negligible addition:

Logistic Model on Headers (Baseline) – Classification Report

	precision	recall	f1-score	support
center	0.42	0.45	0.43	850
left	0.37	0.36	0.37	851
right	0.34	0.32	0.33	851
accuracy			0.38	2552
macro avg	0.38	0.38	0.38	2552
weighted avg	0.38	0.38	0.38	2552

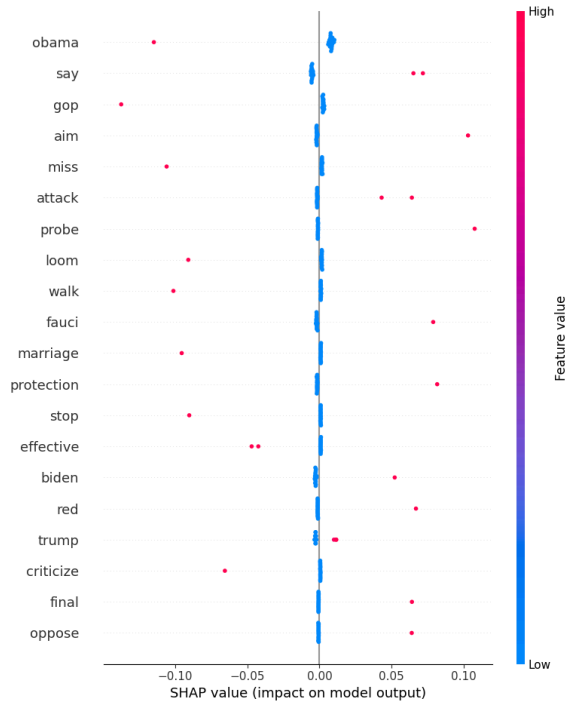
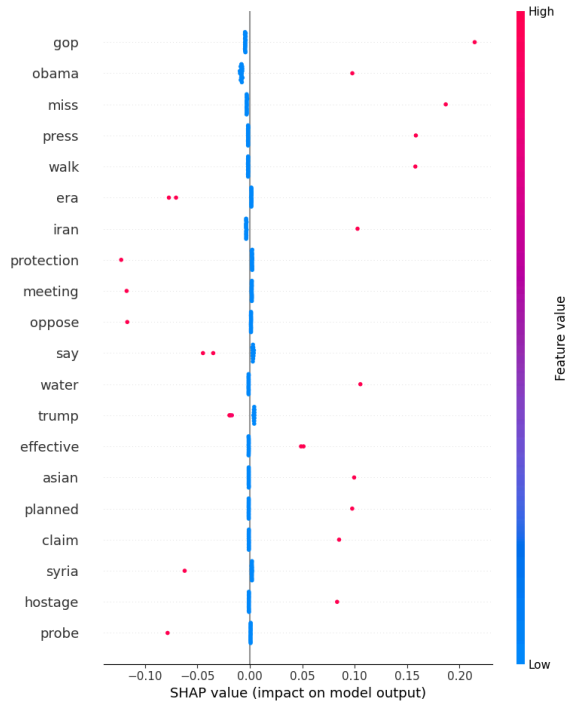
All headline binary classifiers (bias/unbiased) sat at 50-60% accuracy, which is little better than random chance and not much better, relatively, than the C/L/R classifier. This suggests that there isn't a "horseshoe theory" type phenomenon in the underlying data where polarized articles, regardless of direction, tend to use similar, extreme verbiage

The overall accuracy of using a distilBERT model was underwhelming, not a large improvement over a simpler BoW model. Likely because I'm imposing limits on batch size and epochs based on time and computation power

We see shifts in class-wise performance; the baseline model showed more balanced recall across all three classes (left: 36%, center: 45%, right: 32%), whereas DistilBERT tends to predict "right" bias (recall = 73%) but struggled with "left" and "center" (both at 23%). The dataset was balanced between the three categories, so this suggests that the model is able to flag R relatively easily, but struggles to differentiate between C and L. This intuitively follows from the state of modern print newsmedia, which is largely center-left (whereas conservative media tends to be more bold-facedly right wing).

Finally, increase in precision for "left" (44%) and "center" (47%) in DistilBERT suggests that it is making fewer incorrect predictions for those classes, but at the cost of missing many true instances (low recall)

I also computed and visualized SHAP values for the C/L/R transformer model classifier, as a sense check for which features the model is prioritizing. Based on a small sample of the datapoints (n = 20, compute time was a limiting factor), see below the summary plot for left (left) and right (right)-classified texts

SHAP Value Summary Plots	
Left Bias	Right Bias
	

For instance, a mention of “gop” will move the classifier toward Right, while its absence (blue dots) marginally moves the classifier to towards Left, likely because the political left might refer to the same body as “the right” or “republicans” as opposed to the formal party name. Mention

of Fauci leans left, likely because leftist media was more likely to report on him as an authority in 2019-2022. These SHAP values generally seem reasonable for the eclectic sample of 20 headlines, and suggest the classifier is picking up on real features of the text, albeit clumsily.

Ideal Data

Ideally, I would like to have news headline data through to the present day, as opposed to 2022, but AllSides appears to have made the relevant link table difficult to scrape (JavaScript fetches & renders it after the page loads) or to be fetching articles from an API, so there isn't an efficient way to collect large swathes of it. It would be interesting to see the proportion of biased news in the prelude to the 2024 presidential election, in particular.

Furthermore, a lot of Americans get their news through nontraditional—and often more overtly biased—sources (Substack, podcasts, social media), which isn't covered in this dataset.

Extension

A lot of my roadblocks would resolve with more time and compute power, frankly, the foremost being able to fine-tune the distilBERT model with a higher batch size and more epochs. Additionally, even using a colab's TPU runtime (which is itself limited; I had to make several separate accounts to get through the processing I wanted), it takes nearly an hour to produce SHAP values for 20 datapoints. Ideally, I would like to fill out a proper summary plot for n-thousand datapoints to get an accurate understanding of feature importance, and whether the distilBERT model is picking up on correct/relevant information.

I would also delve into the debiasing techniques Baly et al. explores; I spent most of my time focusing on the basics of NLP–being new to it–and didn't have a strong enough understanding of techniques like adversarial adaptation (a media classifier that forces the model to extract bias-relevant information while ignoring media-specific cues), and triple loss pre-training (such that ideologically similar articles are grouped closer together, regardless of their media source) that I felt confident employing them. Additional months would be put towards understanding how these techniques translate to shorter blocks of text and finding the libraries to apply them.

Finally, it would be fun to create a simple GUI that allows users to paste a headline in and see if, per the classifier, it pings as R/L/C biased. That could be accomplished easily in 1-3 months.

Effort

I spent a surprisingly large amount of effort on data collection, in that I was attempting to update QBias' sample web scraper (via Selenium) to build out the AllSides' dataset to 2025. It wound up being several frustrated hours of reading documentation and trying to figure it out ground-up with BeautifulSoup, to no avail.

Thereafter, data processing and tokenization was relatively easy, given the abundance of NLP resources available; as was my baseline logistic regression, RF, and SVM, which mirrored what we did in class. It was specifically the transformer-based models that I struggled with, often running into bugs that I had to pick through the documentation to understand. This section was also full of lengthy run times while training and fine-tuning models, and prompted me to look into batch processing and saving completed models to my device to avoid recomputing them.

Finally, I put a fair amount of effort into visualizing SHAP values and understanding how to interpret summary plots. This was also quite computationally intensive (retrieving values for 50 datapoints, for instance, prompted colab to shut down midway through running the cell) and I had to do some debugging to make sure I was passing the data wrapped in the correct objects/shape, all of which took some weedy library knowledge.

Trade-Offs

The first major tradeoff was between a sufficiently high sample size and the length of text to process. I initially planned to analyze 500-800w length samples of text, which was feasible with the bag-of-words approach, but proved completely unworkable with the transformer-based models (more sophisticated feature extraction)—opting for distilBERT, employing batch processing, and toggling T4/GPU on GoogleColab still resulted in a runtime exceeding 12 hours. The choice was between downsizing the sample size or downsizing the amount of text to process, and because accuracy didn't change too drastically in text vs heading-based classifications for my baseline, BoW model (0.45 versus .38 accuracy), I chose to downsize the amount of text and focus on classifying bias based on news headline alone.

Another back-and-forth was the question of whether to classify on L/C/R or biased (L/R) or unbiased (C). I figured the classifier may struggle to differentiate between the three categories,

with Center as an intermediate. It would not be practical to dispose of C and leave L vs R, for my purposes, as we want to flag work that is tainted with any manner of bias. Thus, I instead framed a binary classifier as biased (L || R) or unbiased (C), as intuitively, any kind of bias may have strong words/similar linguistic features that are not in keeping with the tone of an objective article. I ultimately fit all four models in both settings (see: Final_Project (Tripartite).ipynb and Final_Project (Binary).ipynb). Ultimately, I found the binary classifier was not an improvement, and returned to the three-class model.

Thus, I ultimately selected a three-class model that classified headlines, rather than full length news articles.

Intrigue

This was my first time delving into NLP, so I think it was a particularly applied introduction to the concept as I built out a project from start to finish. I think the different methods I read about for featurizing text were interesting, from simple approaches like the intuitive BoW, to TF-IDF (which complicates things slightly by prioritizing important words over frequent but uninformative words), to transformer-based models with a lot of wacky things going on under the hood.

I think it's also interesting how you can pick a model's brain, so to speak, with SHAP and visualize feature importance in a way that remains intelligible, if only as a sense check. I learned about violin and dot summary plots of SHAP values.

Bibliography

Baly, R., Martino, G. D. S., Glass, J., & Nakov, P. (n.d.). *We can detect your bias: Predicting the political ideology of news articles*. ACL Anthology. https://aclanthology.org/2020.emnlp-main.404/

GeeksforGeeks. (2024, January 3). *Shap : A comprehensive guide to shapley additive explanations*. https://www.geeksforgeeks.org/shap-a-comprehensive-guide-to-shapley-additive-explanations/

Google Cloud Tech. (n.d.). *Transformer models and BERT model: Overview*. YouTube. https://www.youtube.com/watch?v=t45S_MwAcOw&pp=ygUKZGlzdGlsYmVydA%3D%3D

Headline roundups | allsides. (n.d.). https://www.allsides.com/headline-roundups

Irgroup. (n.d.). *Irgroup/qbias: Qbias - A dataset on media bias in search queries and query suggestions*. GitHub. https://github.com/irgroup/Qbias/tree/main

Nadeem, M., & Raza, S. (n.d.). Detecting bias in news articles using NLP models. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1224/reports/custom_116661041.pdf

Sanchez_P, Andrey Popov, & Mario. (1964, November 1). *Correct interpretation of summary_plot shap graph*. Data Science Stack Exchange. https://datascience.stackexchange.com/questions/65795/correct-interpretation-of-summary-plot-shap-graph

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (n.d.). *Distilbert, a distilled version of Bert: Smaller, faster, ...* ResearchGate. https://arxiv.org/pdf/1910.01108