An NLP Pipeline for Bias Detection and Explanation

Giulia Rivetti
s4026543

Kacper Kadziolka
s4115945

Abstract

With the proliferation of digital news, detecting bias in online articles is critical for mitigating misinformation. This work focuses on sentence-level bias detection, leveraging a publicly available dataset that spans multiple news topics. We compare classical machine learning algorithms (Naive Bayes, Random Forest, SVC) and a simple deep neural network trained on various text encodings (Doc2Vec, TF-IDF, Word2Vec, SBERT). Our findings show that a deep learning approach using Doc2Vec embeddings achieves superior accuracy, outperforming a fine-tuned BERT model—likely due to data imbalance and the small size of the dataset. Additionally, we introduce an interpretability pipeline using SHAP, which highlights which words drive a sentence-level bias prediction. Although SHAP explanations at the single-sentence level provide clear insights, aggregating interpretations across the entire test set remains challenging. This study emphasizes the need for balanced data, careful preprocessing, and transparent NLP approaches to effectively detect and explain bias in news articles.

1 Introduction

In today's digital age, accessing news has never been easier, with web-based media becoming the primary channel for gathering information. However, alongside this unparalleled ability to stay updated on nearly any imaginable topic, we are also increasingly exposed to the pervasive presence of bias. This not only impacts our objective understanding but also subtly shapes our opinions, often without us realizing it.

While substantial research has been conducted in the area of automatic bias detection, most existing approaches focus on the document or source level, often overlooking the subtle presence of bias within individual sentences. Another significant limitation highlighted by various researchers is the predominant focus on political datasets, with little exploration of bias across broader news topics.

These challenges have been addressed in a recent study by Lim et al., which developed a dataset featuring sentence-level bias annotations across diverse news topics (Lim et al., 2020). Building upon this resource, we aim to evaluate the efficiency of various natural language processing models, including advanced transformer-based models such as *BERT*, alongside classical machine learning approaches to establish a baseline.

We plan to experiment with the impact of different preprocessing and language filtering methods on the baseline models. Furthermore, we place a strong emphasis on model interpretability. To this end, we propose a novel classification pipeline framework designed to identify specific words and phrases that contribute to a sentence being classified as biased. By integrating tools such as *SHAP* (Lundberg and Lee, 2017a) into our dual pipeline, we not only aim to ensure accurate bias detection but also provide transparent explanations, fostering greater trust and fairness in the model's decision-making processes.

To guide our assignment, we have defined the following research questions:

1. What is the performance of transformer-based models such as *BERT* compared to that of classical machine learning models when detecting sentence-level bias in news articles?

2. How do different preprocessing techniques and encoding methods affect the performance of the selected model?

3. Which parts of a phrase contribute most to a sentence being classified as biased or not?

In the following sections, we will address each of these research questions.

2 Background/related work

With the recent growth of online platforms and their usage, the detection of bias in online news has become a critical area of research in the last years. News articles are not always written in a neutral way, but sometimes the point of view of the writer can arise (Fred Morstatter and Liu, 2018), creating what is defined as media bias. The introduction of unjust subjectivity can be very harmful and therefore detecting bias in news articles is crucial to prevent misinformation (Nadeem and Raza, 2021).

Many studies have been conducted on online bias detection using NLP techniques. Nadeem et al. analyzed various natural language processing (NLP) algorithms in order to build a deeper understanding of the machine learning techniques required to detect biased political leanings in news sources, implementing a deep neural network and a contrastive learning framework called SimCSE (Nadeem and Raza, 2021). Rakhecha et al. provide a comprehensive review of existing studies on online bias detection using NLP. In particular, they examined several techniques, such as data pre-processing, feature extraction, classification and prediction, experimenting with several models, like BERT, LSTMs or Naive Bayes (Khushi Rakhecha and Bhatt, 2023). Besides these works, additional studies have explored explainability in bias detection, with the purpose of being able to interpret the models employed for this task. Ribeiro et al. introduced LIME (Local Interpretable Model-agnostic Explanations), a technique for interpreting model predictions by identifying the most significant features influencing the classification outcomes (Marco Tulio Ribeiro and Guestrin, 2016). Lundberg and Lee proposed SHAP (SHapley Additive exPlanations), a framework for explaining machine learning model outputs based on cooperative game theory (Lundberg and Lee, 2017b). Sallami and Aimeur introduced FairFrame, a framework that detects and mitigates bias in textual data and additionally incorporates LIME to interpret the rationale behind bias detection (Sallami and Aimeur, 2024).

3 Data

For this project, we have employed the dataset proposed by Lim et al. (Lim et al., 2020), which provides sentence-level bias annotations for news articles. The dataset spans four major events-referred in the paper as NFL, FACEBOOK, NORTH KOREA, and JOHNSON-covering news reported between September 2017 and May 2018. The articles were collected from *Google News* and resulting in a dataset characterized by 371 articles for *NFK*, 103 articles for the event *FACEBOOK*, 39 articles for *NORTH KOREA*, and 44 news articles for the event *JOHNSON*.

The dataset presents bias classification both at sentence and article levels, obtained through crowdsourcing. The resulting annotations classify bias into four categories: *Neutral and Not Biased*, *Slightly Biased but Acceptable*, *Biased* and *Very Biased*. An example of some sentences with their labels is presented in Table 1.The distribution of each bias label at sentence level for the four events is reported in Figure 1. By looking at this plot, we can observe that the dataset is unbalanced: not only the distribution of bias labels differs for each event, but also within each event it can be noticed that certain bias categories are more frequent, such as the *Slightly Biased But Acceptable* label. We observed that this distribution could influence the model's predictions, potentially making it biased towards the majority class.

The dataset contains different annotations for each sentence, and since these annotations can differ a lot between one another, we have decided to train our classifiers only using one annotation for each sentence.

4 Methods

We used machine learning classifiers as baseline models, incorporating a manual step to prepare the text by cleaning and encoding sentences. To provide a comparison, we expanded this subsection to include deep learning approaches 4.1. To implement the explainable pipeline discussed in Chapter 4.2, we utilized the Hugging Face library, which provides all the necessary components to build the SHAP pipeline out of the box.

4.1 Naive ML and Deep Learning Classifiers

In order to perform bias sentence classification, the first step that we have followed is to ”clean”

Sentence	Label
A Kentucky lawmaker accused of sexually assaulting a teenage girl in 2013 killed himself Wednesday, officials say, a day after he denied the allegations.	Slighly biased but acceptable.
Shortly before his death, Johnson posted a rambling message on social media, denying the sexual assault allegations and urging his family to stay strong for his wife.	Neutral
Dan is gone but the story of his life is far from over," Rebecca Johnson said through Adams.	Biased
The tumult began Monday, when the Kentucky Center for Investigative Reporting published allegations that Johnson sexually assaulted his daughter's friend during a sleepover in 2013.	Very Biased

Table 1: Example sentences from the considered dataset with one of the labels obtained via the performed crowdsourcing task for each sentence.

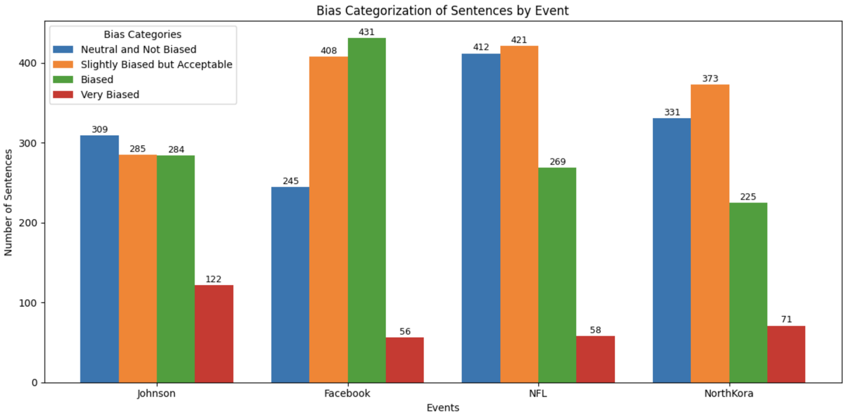


Figure 1: The barplot shows the classification of sentences by bias for each event. The x-axis shows for each of the 4 events, the number of sentences that were classified as *Neutral and Not Biased*, *Slightly Biased but Acceptable*, *Biased* and *Very Biased*.

the data, namely lowercasing, removing punctuation and eliminating stop words from all of the sentences. This preprocessing step is crucial for models based on words as features, because natural language is unstructured and noisy and it needs to be "cleaned" in order to be used by machine learning models (Aydin, 2023). Therefore, we applied this cleaning step for Naive Bayes, Random Forest and SVC, but not for the deep learning model, which is capable of learning complex patterns from raw text and consequently would not benefit from cleaning the text as standard ML models would do.

Before training some models on our data, we have also tokenized the sentences and then applied text encoding, in order to represent the text into vector representation such that it can be used by the model to understand the context of sentences (Bose, 2020). In particular, we have experimented with four different models to encode sentences:

- **Doc2Vec**, which encodes a whole document of text into a vector of chosen size. We have employed two distinct variationa of this model: Distributed memory (PV-DM) and Distributed bag of words (PV-DBOW) (Najkov, 2022).
- **SBERT** or SentenceBERT, which computes sentence embeddings using the BERT model.
- **TF-IDF** (Term Frequency - Inverse Document Frequency encoding), which gives to every word a relative frequency coding with respect to the current sentence and the whole document.
- **Word2Vec**, an encoding system that learns word vectors using a neural network with a single hidden layer.

The next step that we have followed, was to use the encodings that we have obtained for training some machine learning models:

- **Naive Bayes** classifier: this model, which is one of the simplest in machine learning, assumes that all the features are uncorrelated from each other and uses basic probability principles to calculate the result (Najkov, 2022).
- **Random Forest** classifier: it generates several decision trees to classify the input.
- **Support Vector** classifier: this model maximizes the margins between different class values from the data, creating in this way a hyperplane the best divides the dataset (Najkov, 2022).
- **Deep Learning** method: we have employed an artificial neural network with 3 hidden layers and a softmax activation function on the output layer.

4.2 Transformer's interpretability pipeline

To build our NLP explainable pipeline, we decided to use SHAP (Lundberg and Lee, 2017a), which is a game-theoretic approach to explain the output of any machine learning model. The SHAP library is well integrated with the Hugging Face library, providing seamless integration with various pre-trained models. Therefore, for the interpretable pipeline, we chose to use Transformer models within a Hugging Face environment. We further narrowed our choice to a *BertForSequenceClassification* pre-trained model, suitable for supplementary fine-tuning for the specific task we are working on. To proceed with the task at hand, we followed the standard procedure of fine-tuning the Hugging Face model by fetching the tokenizer and the model itself. First, the dataset was tokenized and mapped to the required format, followed by actual fine-tuning on the task. The experimentation was conducted with two distinct approaches to the problem 5.2, we ran the learning sequence for both the classification and ordinal regression configurations.

Once the fine-tuned model was ready to make predictions, we began to form the explainable pipeline. Fully aware that ordinal regression was a more suitable candidate for our problem, we decided to proceed with a 4-class classification model type. This decision was due to Hugging Face's capability to be integrated with SHAP, thereby allowing us to meet the required implementation timeline. We used the Hugging Face Transformer's pipeline, specifically the *TextClassificationPipeline*. The pipeline object accepts the model and tokenizer as initialization parameters. The fitted pipeline object is subsequently passed to the explainer instance for SHAP predictions. The explainer object is later reused for calculating SHAP values and computing various plots based on the provided textual input.

5 Results

We made the following distinctions. First, we trained naive classification models preceded by the manual cleaning and encoding of sentences 5.1. Second, we experimented with BERT-based models for the sequence classification task 5.2. Finally, we introduced an explainable pipeline using SHAP 5.3, which helped us understand the rationale behind Transformer-based classification by highlighting specific keywords that contributed to biased labels, allowing us to draw some interesting conclusions.

5.1 Encoded Sentences Classifiers

Table 2 shows the accuracy obtained on the test set by each combination of encoder-classifier that we have mentioned earlier. By looking at the table we can see that the best performance is achieved when using Doc2Vec (PV-DM) as encoding technique for each classifier. Indeed Doc2Vec is particularly effective at capturing contextual relationships within sentences, making it the best encoding technique in our case. While TF-IDF only relies on word frequency and Word2Vec is based on word-level embeddings without aggregating sentence-level meaning, Doc2Vec creates dense vector representations that reflect both the semantic meaning of individual words and their relationships within the sentence. SBERT, though generally powerful, is not fine-tuned for our particular task and therefore leads to lower performance.

Since the accuracy is not the only metric that should be taken into account when comparing models, we have reported in Table 3 the performance of the models using Doc2Vec (PV-DM) as encoder across several metrics: accuracy, precision, recall and F1 score. By looking at the ta-

	Doc2Vec(PV-DBOW)	**Doc2Vec(PV-DM)**	**SBERT**	**TF-IDF**	**Word2Vec**
Naive Bayes	0.375	0.546	0.415	0.295	0.265
Random Forest	0.33	0.746	0.4	0.4	0.405
SVC	0.395	0.631	0.405	0.425	0.38
Deep Learning	0.751	0.852	0.796	0.796	0.790

Table 2: Table showing the accuracy obtained on the test using a combination of each encoder and classifier type that we have considered.

ble it can be observed that the accuracy and the recall show the same values for each classifier; this happens because the recall was computed as micro-averaging and in multi-class classification micro-recall coincides with the accuracy (Learn, 2024). The table shows that among the classifiers, the deep learning model achieves the best performance across all metrics, with an accuracy of about 85%, which highlights the ability of the deep learning model to capture complex patterns in the data. Moreover, it achieved high performance also in terms of precision and recall, reflecting its ability to both correctly identify biased sentences and minimize false positives and false negatives. The Random Forest classifier also shows good results, thanks to its ability in handling complex, non-linear relationships in data. SVC and Naive Bayes, which are less sophisticated models, are not well suited for our task as the results in the table suggest.

5.2 BERT Family

Using Transformer models for this task remains a significant challenge in our analysis. We noticed that after further preprocessing the dataset, the data capacity shrank by a factor of four. This affected the generalizability of the Transformer model, which continued to overfit the training data. The model converged to the optimum on the training dataset but failed to achieve more than 50% accuracy on both the validation and test sets.

To further experiment, we redefined the problem as an ordinal regression configuration. The *CrossEntropy* loss was switched to *MSE*, the number of labels was reduced from four to one, and the labels were cast as float values. This allowed us to train BERT as a regression problem. However, we observed the same pattern of overfitting the training data.

To mitigate overfitting, various regularization techniques could be implemented. However, in our work we continue to compare different classifiers 5.1 and BERT family models with an objective of our explainable pipeline 5.3.

5.3 SHAP - Explainable Pipeline

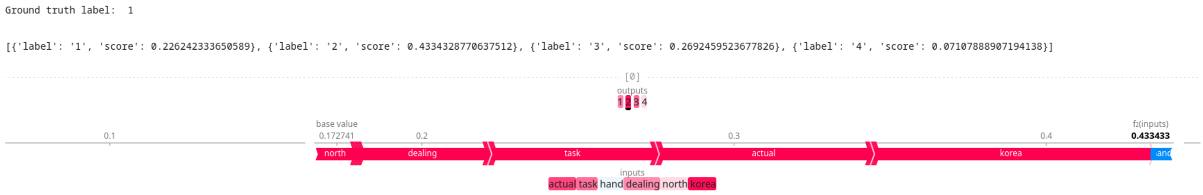
The SHAP plots helped us understand two major factors related to Transformer classifiers. The first is the weighting of words in single-sentence classification, particularly what causes the model to label a given sentence as belonging to a specific class. The second is plotting the top words that impact a given class, following the tutorial from the SHAP documentation (Docs, 2024), thus providing us with the most influential words across the entire test set.

We randomly sampled three sentences from the test set to run through our explainable pipeline. As a result, we obtained more detailed information about the total weight scored by each of the four possible classes, as well as the contribution of each word toward a given class. Figure 2 represents a visualization of the impact on class 2 for the following text: 'actual task hand dealing North Korea'. We can observe the confidence levels for each of the assigned classes: approximately 0.23, 0.43, 0.27, and 0.07 for classes 1, 2, 3, and 4 respectively. Although our model misclassified the ground truth label as class 1, assigning it roughly 0.23 probability, it is interesting to see which words led the model to classify this sentence as class 2: 'Slightly Biased but Acceptable'. Despite the fact that the human annotator marked this sentence as unbiased, our fine-tuned model identified the word 'North Korea' as the most important factor in classifying this sentence as slightly biased. We are aware that this is more about the sequence of words (the whole sentence), but it is highly interesting to see that our model somehow associated 'North Korea' with a slight bias, which is not entirely surprising given the country's political stance.

Further experimentation with SHAP features led us to plot the top words impacting specific

	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.5458	0.6308	0.5458	0.5619
Random Forest	0.74583	0.7536	0.7458	0.7348
SVC	0.6306	0.5943	0.6306	0.6038
Deep Learning	0.8517	0.8672	0.8761	0.8716
BERT	0.4260	0.3104	0.4260	0.3581

Table 3: Table comparing the results obtained after training using Doc2Vec (PV-DM) as encoding technique. The table reports accuracy, precision, recall and F1 score for the four distinct classifiers we have employed.



```
Ground truth label:  1

[{'label': '1', 'score': 0.226242333650589}, {'label': '2', 'score': 0.4334328770637512}, {'label': '3', 'score': 0.2692459523677826}, {'label': '4', 'score': 0.07107888907194138}]
```

Figure 2: The SHAP values and the model confidence computed for a text 'actual task hand dealing north korea'.

classes. We started by extracting all the events concerning North Korea. We used a previously prepared explainer pipeline to compute the SHAP values for all the test events of the filtered category. This way, we prepared a dense representation of word features for all test examples of a single category. We were asking ourselves if we could identify a single word or a couple of words that are the most biased or the least biased (i.e., the most neutral words from a text corpus).

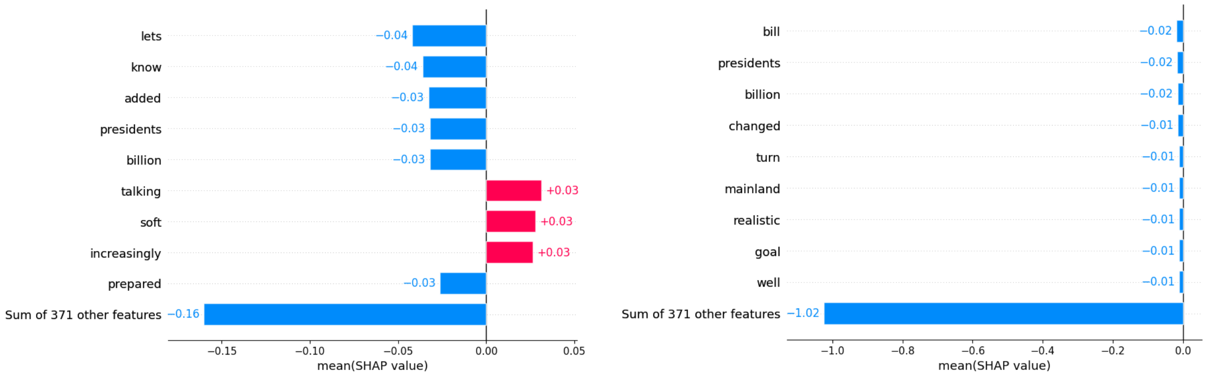
Results can be seen in Figure 3. We provided separate plots for a neutral, unbiased class and one explicitly for the most biased text samples. We notice a significant difference in the importance of the top words in both cases. By studying the neutral plot 3a, we see that words such as "talking," "soft," and "increasingly" contribute to the output, while words like "lets" or "know" have a negative impact on the first class. On the other hand, the importance shown in the second plot 3b does not reveal any information from which we could draw a conclusion. We believe that either the bias in sentences is highly dependent on the sequence as a whole (rather than on single words), or we observe an imbalance between the neutral and highly biased classes, as was observed in the distribution plot 1.

6 Discussion

If we look back at the results presented in the previous section, we can observe that BERT, despite being a state-of-the-art model, in this case achieved lower performance compared to simpler models like Deep Learning and traditional ML models. The dataset that we have considered for this assignment, as showed in Section 3 is highly unbalanced and BERT performance can be affected by this; to address this issue, BERT should be meticulously fine-tuned, which is not always an easy task. Moreover, BERT performs better with a larger dataset and for this task it would have been therefore beneficial to have more training samples.

As we had anticipated while looking at the dataset in Section 3, the dataset imbalance poses a significant challenge for this task. This issue could be identified as the cause of BERT's performance, as the model might have been inclined to predict the majority class, evincing in overfitting training data. Simpler models like Random Forest and Naive Bayes are typically less affected by such imbalances, leading to higher performance in this case.

Our explainable pipeline works well with single-sentence classification, clearly providing the rationale behind model classifications and each word's contribution to the final softmax score across the model's output head. However, the bar plot representing top words across the entire test set 3 has an ambiguous meaning due to the curse of dimensionality of the word features. We have not explored computing the SHAP values and therefore providing an explainable pipeline for ordinal



(a) Impact on 1 class - (*Neutral and Not Biased*).

(b) Impact on 4 class - (*Very Biased*).

Figure 3: SHAP bar plot explaining the top words impacting a neutral and very biased labels.

regression or baseline machine/deep learning approaches 5.1.

7 Conclusion

In conclusion, our experiments highlight that traditional machine learning and simple deep learning models—particularly when combined with Doc2Vec encoding can outperform BERT-based models in sentence-level bias detection under conditions of limited and imbalanced data 5.1. While BERT remains a state-of-the-art approach, this study underscores the importance of robust preprocessing, careful fine-tuning, and balanced, wide dataset for optimal performance 5.2. Furthermore, the SHAP-based interpretability pipeline provided valuable insights into how specific words contribute to bias predictions, particularly at the single-sentence level 5.3. However, interpreting aggregated results across the entire test set remains challenging due to the curse of dimensionality in language features. Future work could explore advanced regularization and data augmentation strategies to address model overfitting, as well as extend the explainable pipeline to ordinal regression and classical machine learning baselines for a more comprehensive bias-detection framework 6.

Contributions of the team members

Giulia - code (dataset preparation and analysis, encoding methods, ML and deep learning classifiers); report (background, data, methods, results, discussion)
Kacper - code (BERT, SHAP, explainable pipeline implementation) ; report (abstract, introduction, methods, results, discussion, conclusion)

References

[Aydin2023] Aysel Aydin. 2023. 1 — Text Preprocessing Techniques for NLP.

[Bose2020] Bishal Bose. 2020. NLP — Text Encoding: A Beginner's Guide.

[Docs2024] SHAP Docs. 2024. Emotion classification multiclass example.

[Fred Morstatter and Liu2018] Uraz Yavanoglu Stephen R. Corman Fred Morstatter, Liang Wu and Huan Liu. 2018. Identifying framing bias in online news. *ACM Transactions on Social Computing 1*, pages 1–18.

[Khushi Rakhecha and Bhatt2023] Muskan Agrawal Khushi Rakhecha, Simran Rauniyar and Aruna Bhatt. 2023. A survey on bias detection in online news using deep learning. *2nd international conference on applied artificial intelligence and computing (ICAAIC)*, page 396–403.

[Learn2024] Scikit Learn. 2024. classification$_{r}eport.$

[Lim et al.2020] Sora Lim, Adam Jatowt, Michael Färber, and Masatoshi Yoshikawa. 2020. Annotating and analyzing biased sentences in news articles using crowdsourcing. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1478–1484, Marseille, France, May. European Language Resources Association.

[Lundberg and Lee2017a] Scott Lundberg and Su-In Lee. 2017a. A unified approach to interpreting model predictions.

[Lundberg and Lee2017b] Scott Lundberg and Su-In Lee. 2017b. A unified approach to interpreting model predictions. *NeurIPS (Long Beach, California, USA)*, pages 4768–4777.

[Marco Tulio Ribeiro and Guestrin2016] Sameer Singh Marco Tulio Ribeiro and Carlos Guestrin. 2016. ”why should i trust you?”: Explaining the predictions of any classifier. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM*, pages 1135–1144.

[Nadeem and Raza2021] Muhammad Umar Nadeem and Sarah Raza. 2021. Detecting bias in news articles using nlp model. *Stanford CS224N Custom Project*.

[Najkov2022] Danilo Najkov. 2022. Detecting political bias in online articles using NLP and classification models.

[Sallami and Aimeur2024] Dorsaf Sallami and Esma Aimeur. 2024. Fairframe: a fairness framework for bias detection and mitigation in news. *ACM Transactions on Social Computing 1*.