Unveiling Bias in Political Discourse Using NLP

Abdulaziz Houtari and **Avi Lochab** and **Samir Shakir** and **Simon Situ**
Computer Science Department
Michigan State University

Abstract

In this paper, we investigate the detection of political bias in Western media sources using Natural Language Processing (NLP) and Machine Learning (ML) models. Our study initially focused on implementing deep learning techniques to automatically detect bias in media articles from datasets labeled across a political spectrum. Initial experiments with GPT-4 and a custom CNN-GRU model on the MBIC dataset demonstrated significant challenges. The GPT-4-based model achieved only 22% accuracy, while the CNN-GRU model attained around 50-55% accuracy but exhibited signs of overfitting, with training accuracy reaching 100% and validation accuracy stagnating around 55-60%. To address these issues, we plan to explore advanced techniques such as data augmentation and LLM-based prompting to improve classification. This report expands upon our methodology and outlines planned experiments for further improving model performance.

1 Introduction

Media outlets and other sources of information are playing an increasingly important role in informing the public of current events. However, information is often distributed with varying degrees of bias, influenced by political, social, or economic agendas. Bias in media can distort facts according to the source's views, which can then influence the audience's views and behavior if the audience is not aware of the biases. Detecting such biases can be useful for the public to separate the facts from biased content. While traditional natural language processing (NLP) techniques have been applied to identify bias by focusing on linguistic features such as word frequency and sentiment, they are prone to modeling the topics, writing styles, and the source of the media instead of the underlying bias [6, 7].

With advancements in deep learning, more sophisticated models have been proposed and adopted to automatically detect bias given large datasets. These models can analyze context and semantics more effectively, leading to higher accuracy compared to traditional methods. This project aims to compare advancements made to tackle the difficulties of bias detection and implement a state-of-the-art deep learning model for the task of detecting bias in journalistic content, focusing on media articles from diverse political and ideological backgrounds. By replicating an existing model, our goal is to evaluate its performance in identifying bias across news outlets, contributing to the ongoing efforts to ensure accountability in media.

2 Related Works

Research in bias detection has been approached from two main perspectives: one using pure natural language processing (NLP) techniques, and the other utilizing machine learning and deep learning models. Both approaches have their own advantages and disadvantages in the task in question.

2.1 Traditional NLP Techniques

One approach to detecting bias in media articles involves traditional NLP techniques, which rely on linguistic features such as word frequency, cosine similarity, and syntactic patterns. In Baraniak and Sydow's study, methods such as TF-IDF, keyword analysis, and document similarity (using cosine distance) were employed to detect bias based on news article similarity (Baraniak and Sydow, 2018). These methods were shown to be effective in grouping articles on the same event, highlighting potential biases through comparative analysis of different sources.

Traditional NLP techniques, such as TF-IDF, cosine similarity, and keyword analysis, are often used to detect bias through article similarity. These methods are computationally efficient and provide interpretable results, making them suitable for comparing news sources. However, their reliance on surface-level features limits their ability to capture deeper, contextual nuances of bias. They often struggle with more complex forms of bias, such as framing or tone, and may require manual tuning for effective performance (Garrido-Muñoz et al., 2021).

2.2 Supervised Learning Approaches

Supervised learning models, for example, Support Vector Machines (SVM), and logistic regression are trained on labeled datasets to identify biased language. In the context of media bias detection, these models are typically trained on a dataset where articles or statements are pre-labeled as biased or unbiased. For example, a study by Mahanta utilized a logistic regression model combined with linguistic features like word embeddings to detect bias in American Media [6]. While these models can achieve high accuracy with sufficient labeled data, they are heavily dependent on the quality and quantity of labeled examples. Moreover, they may struggle to generalize across different topics or sources if the training data is not representative.

2.3 Unsupervised and Semi-supervised Learning Techniques

In cases where labeled data is scarce, unsupervised and semi-supervised learning methods offer alternative approaches. This includes techniques such as clustering, topic modeling, and autoencoders. Additionally, a study by Kulkarni et al. explored the use of Latent Dirichlet Allocation (LDA) to identify thematic biases across a corpus of news articles, finding that certain topics were consistently underrepresented in some media outlets [7]. These methods are useful for exploratory analysis and can reveal subtle biases without requiring labeled data. However, their interpretability and performance can vary significantly depending on the chosen model and the nature of the dataset.

2.4 Deep Learning Models

Deep learning models, particularly those based on neural networks, have demonstrated superior performance in capturing the nuanced semantics of language. For example, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) networks, have been employed for media bias detection. These models can automatically learn hierarchical representations of text, making them effective at identifying complex patterns of bias that are not evident through traditional methods. A notable study by Chumachenko utilized a Bidirectional LSTM network to detect bias in news articles by analyzing the sequence of words and their contextual meanings, achieving state-of-the-art results in various bias detection tasks (Padalko et al., 2024). Despite their high performance, deep learning models require large datasets and substantial computational resources for training, and they often function as black boxes, making their predictions difficult to interpret.

2.5 CNN-GRU Models

Sharma et al. proposed an ensemble model that combines the capabilities of a 1-D Convolutional Neural Network (CNN) and a Bidirectional Gated Recurrent Unit (GRU) model. The CNN model captures local patterns within sequences while the GRU can capture contextual information and long-range dependencies (Sharma et al., 2023). The outputs from both models are concatenated and given to a fully-connected layer which will make the final probability distribution over the possible classes. They found that the model with pre-trained GloVe embeddings outperforms other methods on the NewB and MBIC datasets such as SVM, Naive Bayes, Random Forest, and Feed Forward Neural Networks when paired with embeddings such as TF-IDF, Word2Vec, and BERT.

3 Methodology

In this project, we planned to detect media bias using deep learning models, with a focus on replicating and enhancing the performance of existing methods. We used the NewB and MBIC datasets, from the study by Sharma et al. (2023), as our primary datasets for bias classification. The methodology will proceed as follows:

3.1 Data Collection

NewB dataset (Wei, 2020) includes approximately 200,000 sentences from news articles about Donald Trump, published by 11 media outlets identified as either liberal, neutral, or conservative. We intend to use the political orientation of the media houses to label the articles, allowing for nuanced bias classification.

MBIC dataset (Spinde et al., 2021) comprises articles from major outlets such as The Wall Street Journal, Fox News, The New York Times, and CNN, labeled across a political spectrum from "left" to "right" We plan to use the labels provided by annotators to detect bias at both word and sentence levels. This dataset also labels each article as "biased" or "not biased" based on specific keywords and based on the annotator. We plan on detecting and classifying that kind of bias too in addition to the political bias type.

3.2 Data Pre-processing

Our data pre-processing is slightly different given the model we are experimenting on, however, there are few key steps that we did across all the models:

- **Tokenization**: This step involves splitting the text into individual words or tokens. Tokenization helps in converting the raw text into a structured format where each word can be separately analyzed or used as input for the model. We initially lowercase all the words before this step takes effect. This step differed slightly in-between models; we used the BERT tokenizer imported from HuggingFace's transformers package in Python whereas when we experimented on neural networks we used the NLTK tokenizer.

- **Building a Vocabulary**: Here we just built a vocabulary given the dataset after tokenization to get a clear picture of how big the dataset actually is and diverse it is.

- **Removing stop words**: Stop words are common words (e.g., "the," "and," "is") that do not carry significant meaning and can be removed to reduce the dimensionality of the dataset and improve computational efficiency without losing meaningful information.

- **Stemming and Lemmatization**: These processes help in reducing words to their root forms. Stemming cuts off prefixes or suffixes to get the base form of words, while lemmatization uses linguistic rules to map words to their base or dictionary form. This step helps in normalizing the text and reducing the variability of words.

3.3 Models

We experimented on a range of models, starting with transformers like BERT to get a baseline, and evaluating more complex models at each step until we reach the Bi-GRU CNN model proposed by Sharma et al. (2023).

3.3.1 Plain BERT

We used the `bert-base-uncased` model for sequence classification on the MBIC dataset. BERT (Bidirectional Encoder Representations from Transformers) is a Transformer-based model that reads text bidirectionally to capture context. Pre-trained on large text corpora, it is fine-tuned for specific tasks like bias classification in this case.

The dataset was tokenized using BERT's tokenizer with a maximum length of 128 tokens. We fine-tuned BERT using the `BertForSequenceClassification` model, which adds a classification layer on top of BERT's output. The model was trained for 5 epochs with a learning rate of $2e{-}5$, using the AdamW optimizer and cross-entropy loss. During training, accuracy and loss were monitored, and the best model was saved based on validation performance.

The final model achieved 62.03% accuracy on the validation set, demonstrating its effectiveness in detecting bias in the MBIC dataset.

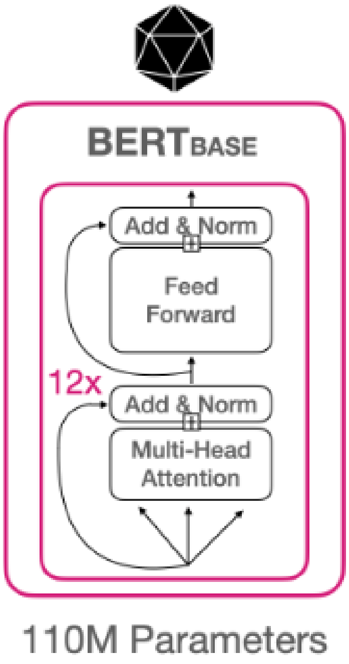

Figure 1: Base BERT

3.3.2 Bert as an Embedding Layer to a Neural Network

The model utilizes a pre-trained BERT model (`bert-base-uncased`) as the core component to generate contextualized embeddings from input text. On top of BERT, a dropout layer with a dropout rate of 0.3 is applied to prevent overfitting during training. This is followed by a fully connected (dense) layer with an input dimension of 768, which outputs logits for classification into 3 classes.

The training process employs the AdamW optimizer, using a learning rate of $2e-5$. The cross-entropy loss function is applied, as this is a multi-class classification task. During each epoch, the model processes batches of size 32, performing forward passes through the BERT model, dropout layer, and the classifier, followed by backpropagation to adjust the model's parameters.

For evaluation, the model is validated on a separate dataset after each epoch, monitoring both the validation loss and accuracy. The model is trained for 3 epochs. These hyperparameters, including the dropout rate, learning rate, batch size, and number of epochs, are carefully tuned to ensure the model converges effectively without overfitting while achieving good performance on the validation set.

3.3.3 CNN-GRU

Sharma et al. proposed an ensemble model consisting of a 1-Dimensional Convolutional Neural Network (1-D CNN) and a neural network with Bidirectional Gated Recurrent Units (Bi-GRU) (Sharma et al., 2023). The model begins with an embedding layer that uses pre-trained GloVe embeddings, which convert input words into embedding vectors. The embeddings capture semantic meanings, and they are trainable to allow fine-tuning during training. The CNN component, which includes 100 filters with a kernel size of 3, detects local patterns in the data, such as important n-grams or phrases. The output of the CNN is passed through a max-pooling layer to reduce its dimensionality, followed by two fully connected (dense) layers for the final class predictions.

The Bi-GRU component, with 64 hidden units for each direction, captures both forward and backward dependencies in the text, preserving the sequential nature of the data. By processing the text in both directions, Bi-GRU provides a comprehensive understanding of the context. This is followed by a dense layer to produce the final class predictions.

The predictions from the CNN and Bi-GRU models are concatenated and passed through a dense layer to perform the final classification into one of the three target classes.

We modified it slightly and simplified the model as can be seen in fig. 2

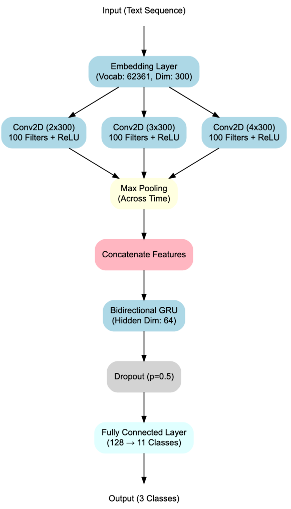

Figure 2: Proposed model on NewB

4 Experiments

In this section, we present how we conducted our experiments. We focused broadly on two tasks: binary bias detection and political bias detection (Multi-class). The experiments were designed to assess the models' performance under different classification tasks, using the MBIC and NewB datasets as described in Section 3.

4.1 Binary Bias Detection

For our first experiment, we concentrated on the binary classification task of distinguishing between *biased* and *non-biased* articles. This served as a baseline to understand the models' capabilities in detecting bias at a fundamental level.

We utilized the MBIC dataset for this task, using the binary labels provided. The dataset was preprocessed as outlined in Section 3.2.

We evaluated three models:

1. **BERT (bert-base-uncased):** Fine-tuned for binary classification.

2. **BERT as an Embedding Layer:** BERT used to generate contextual embeddings, followed by a neural network classifier.

3. **CNN-GRU:** The model described in Section 3.3.3, adapted for binary classification by adjusting the output layer.

The models were trained using appropriate hyperparameters, with early stopping employed to prevent overfitting. We evaluated their performance using metrics such as accuracy, precision, recall, and F1-score. The results are presented in Section 5.

4.2 Political Bias Detection

Our second experiment focused on multi-class classification to identify articles' bias types across multiple labels. This task is the main focus of our paper, aiming to evaluate the models' ability to detect nuanced political biases in media articles.

We conducted experiments on both the MBIC and NewB datasets, as follows:

4.2.1 MBIC Dataset

We utilized the MBIC dataset for multi-class classification, aiming to classify articles into three bias categories: *left*, *center*, and *right*. The dataset was preprocessed similarly to the binary task.

We evaluated the same three models as before, adapted for three-class classification:

1. **BERT (bert-base-uncased):** Fine-tuned for three-class classification.

2. **BERT as an Embedding Layer:** BERT embeddings fed into a neural network classifier with three output classes.

3. **CNN-GRU:** Adjusted to handle three output classes corresponding to the bias categories.

4.2.2 NewB Dataset

For the NewB dataset, we performed experiments in two different ways:

Three-Label Classification In the first approach, we mapped each article to one of three political bias labels (*liberal*, *neutral*, *conservative*) based on the known orientation of its news source. We then trained the models to classify the articles into these three categories.

Eleven-Label Classification In the second approach, we trained the models to classify articles into the original eleven news source labels provided in the NewB dataset. This approach allowed us to assess the models' ability to distinguish between individual news outlets, capturing more nuanced patterns in the data. We did not map the eleven labels back to three labels; instead, we evaluated the models directly on the eleven-class classification task. And then we also only evaluated the model on the mapped labels.

4.2.3 Models and Training

For both approaches, we utilized the same models as before:

1. **BERT (bert-base-uncased):** Fine-tuned for multi-class classification (three or eleven classes).

2. **BERT as an Embedding Layer:** BERT embeddings used in a neural network classifier with the appropriate number of output classes.

3. **CNN-GRU:** Adjusted to handle the number of output classes corresponding to the task (three or eleven).

The datasets were preprocessed as described in Section 3.2. The models were trained using cross-entropy loss for multi-class classification, with appropriate learning rates and optimizers selected based on preliminary experiments. We employed early stopping and regularization techniques to prevent overfitting.

4.2.4 Evaluation

We evaluated the models using metrics suitable for multi-class classification, including accuracy, precision, recall, and macro-averaged F1-score. For the eleven-label classification task, we paid particular attention to the confusion matrix to analyze how well the models distinguished between the different news sources.

We also analyzed the models' performance in terms of their ability to capture the nuanced differences between the classes, especially in the eleven-label classification task. The results and detailed analysis are presented in Section 5.

5 Results

5.1 MBIC Bias Detection: Biased or Not Biased (2 Labels)

Our initial experiments concentrated on binary classification, classifying between biased and non-biased articles. Three distinct models were employed:

1. **BERT (bert-base-uncased):** This model achieved a validation loss of 0.7924 and a validation accuracy of 73.10% after 5 epochs of training, with a total training and evaluation time of approximately 1 minute.

2. **BERT as an Embedding Layer:** By using BERT to generate contextual embeddings, we observed an improvement in performance. Validation

loss decreased to 0.5252, with a validation accuracy of 74.68%, again after 5 training epochs.

3. **CNN-GRU Model:** We used our model proposed in 3.3.3, but slightly modified to fit with the MBIC dataset by modifying the vocabulary size. We kept the parameters for the architecture the same as described by Sharma et al. (embedding dimension of 300 for a vocabulary size of 1,193,514, 100 CNN filters, and a total Bi-GRU hidden dimension of 128). We varied the dropout and optimizer parameters to find the best-performing model on the validation set. With the AdamW optimizer with learning rate 0.001 and weight decay $1e{-}6$, we kept the model parameters that achieved the highest validation accuracy. Evaluating the model on the test set, the model attained an accuracy of 69.87%.

Model	Accuracy
Base BERT	73.10%
BERT Embeddings	74.68%
CNN+GRU	69.87%

Table 1: Testing metrics for the models trained for the MBIC binary task.

5.2 MBIC Bias Type Detection: 3 Labels

The results were as follows:

1. **GPT-4o:** This model demonstrated limited effectiveness, achieving an accuracy of only 22%.

2. **Plain BERT:** Validation performance was subpar, with a validation loss of 0.8972 and validation accuracy of 52.22% after training for 3 epochs.

3. **BERT as an Embedding Layer:** Akin to the binary classification task, this model struggled, obtaining a validation loss of 0.9719 and an accuracy of 64.24% after 3 epochs.

4. **CNN-GRU Model:** This model used a similar architecture as previously described but was adjusted for three output classes.

Training accuracy reached 91.63%, while validation accuracy was much lower at 54.29%. On the test set, the model yielded an accuracy of 48.72%, with a precision of 0.4850, a recall of 0.4872, and an F1 score of 0.4846.

Model	Accuracy
Base BERT	52.22%
BERT Embeddings	64.24%
CNN+GRU	48.72%

Table 2: Testing metrics for the models trained for the MBIC 3-class bias detection task.

5.3 NewB Bias Detection

The results here were promising.

5.3.1 Three-Label Classification

Our proposed model trained on the mapped political labels performed surprisingly well achieving an accuracy of 56% which is a ≈ 2.5% increase in accuracy from the model proposed in Sharma et al. (2023) for the same dataset which only achieved a 53.6%.

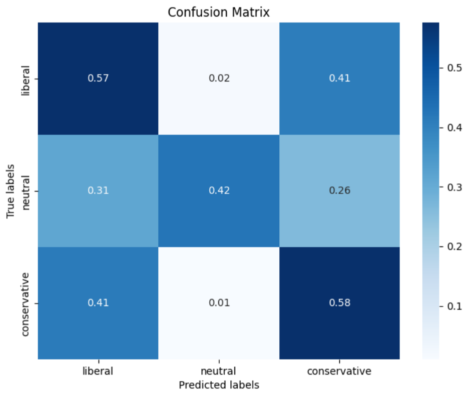

Figure 3: Confusion matrix of predicted an ground truth political bias for our CNN+GRU model trained on three labels

The model did surprisingly well surpassing the other model, but the imbalance in classifying neutral texts can be visibly seen due to the mapping of text source to political label like we mentioned in Wei (2020) and figure 5.

5.3.2 Eleven-Label Classification

When we trained our model on the original 11-labels of the NewB dataset, it performed way better achieving an accuracy on the mapped labels of 62% which is a huge improvement ≈ 9% increase from the model proposed in Sharma et al. (2023) and a ≈ 6% increase from the same model trained on the mapped labels.

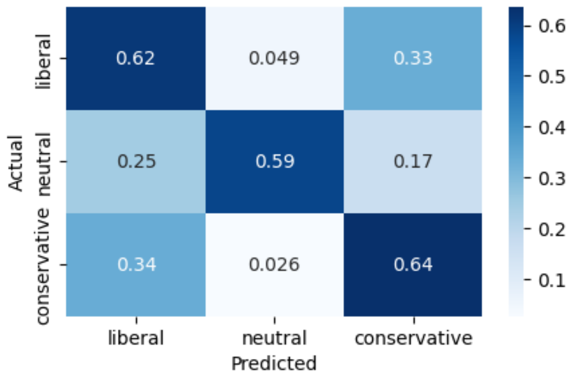

Figure 4: Confusion matrix of predicted and ground truth political bias for our CNN+GRU model trained on the original eleven labels

We can already see big improvements in classifying neutral texts because the class imbalance mentioned before is gone now when trained on all original labels like we can see in figure 5 it evens out all classes.

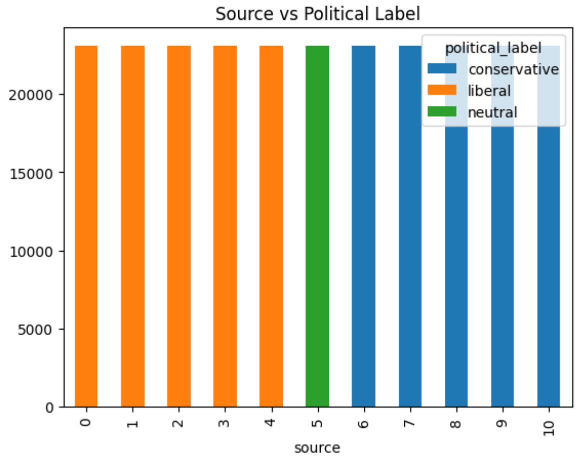

Figure 5: Source vs. Mapped Political Label Distribution of NewB dataset

To also see how this model compares to the model in Wei (2020), we evaluated it on classifying the 11 labels which can be seen in figure 6. Our model achieved an accuracy of 39% compared to the 34% achieved in their paper giving us a $\approx$ 5% increase in accuracy.

5.3.3 Five-gram Analysis

To better see how our model performs, we did a five-gram analysis by giving it 25 five-grams for both liberal and conservative classes. Example of such five-grams are the following. trump has a history of, the trump campaign declined to comment, trump as commander in chief, and trump strengthens the u.s. economy where liberal five-grams are written in blue and conservative five-grams are written in red. The results of this analysis can be seen in figure 7 and are very promising achieving identical accuracy in classifying both liberal and conservative five-grams.

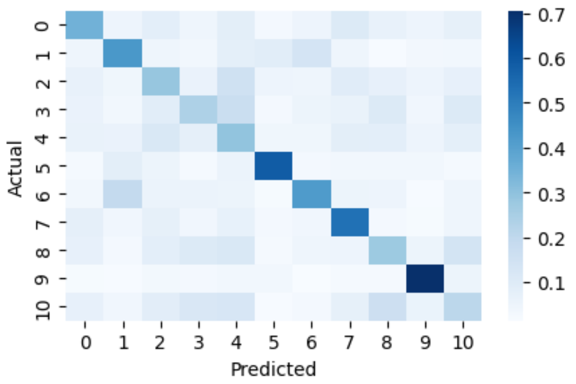

Figure 6: Confusion matrix of predicted and ground truth news sources for our CNN+GRU model

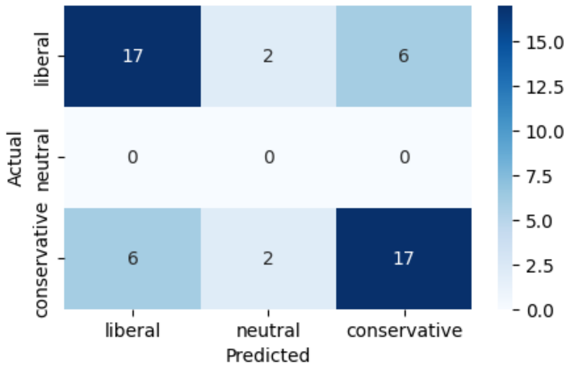

Figure 7: Our CNN+GRU model predicted labels for most significant liberal and conservative five-grams.

6 Conclusion and Future Work

In this study, we investigated the challenging task of detecting political bias in media articles using Natural Language Processing and various deep learning techniques. Through the implementation and evaluation of various models, including traditional approaches like BERT and advanced architectures such as CNN-GRU, we observed notable differences in performance across datasets and classification tasks.

For the MBIC dataset, the CNN-GRU model exhibited moderate performance, achieving higher accuracy for binary classification tasks compared to three-label classification. This highlights the difficulty of distinguishing between fine-grained

Model	Embedding	Accuracy (%)
CNN+GRU Sharma et al. (2023)	Pre-trained GloVE	53.6
Our CNN+GRU 5.3.1	Pre-trained GloVE	56.4
Our CNN+GRU 5.3.2	Pre-trained GloVE	62.2

Table 3: Comparison of results between the different models and different training techniques

bias categories. On the NewB dataset, training with original news source labels and then mapping to the political spectrum demonstrated improved performance, with the CNN-GRU model achieving 62.2% test accuracy. These findings emphasize the importance of data representation in influencing model outcomes.

Despite these advancements, challenges such as overfitting, limited computational resources, and the inherent bias within datasets remain significant barriers to achieving state-of-the-art performance. Additionally, models like CNN-GRU and BERT, while effective, often function as black-box systems, limiting interpretability and hindering real-world applications.

6.1 Future Work

To address the limitations encountered in our project and to further advance our study, we propose to work on the following directions:

- **Dataset Expansion and Balancing:** Collecting larger and more balanced datasets, including articles from underrepresented media outlets, could enhance model generalization and improve classification accuracy across all bias categories.

- **Advanced Architectures:** Incorporating transformer based models such as RoBERTa or fine-tuning domain-specific models could also help improve performance while maintaining interpretability through attention mechanisms.

- **Domain Adaptation:** Experimenting with domain adaptation techniques to better handle bias detection across different contexts or languages could also help improve performance.

- **Model Interpretability:** Exploring methods like SHAP or attention visualization could also help provide insights into how models detect bias, aiding in both transparency and accountability.

- **Ensemble Learning:** Combining outputs from multiple models, such as CNN-GRU and transformers, might improve robustness and accuracy by leveraging strengths of each architecture.

- **Longitudinal Bias Detection:** Developing models capable of detecting sway in bias over time by incorporating temporal data, such as publication dates or historical context.

By addressing these areas, we hope to build more accurate, interpretable, and robust systems for bias detection, contributing to a better understanding of media content and fostering informed decision-making among audiences.

References

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Katarzyna Baraniak and Marcin Sydow. 2018. News articles similarity for automatic media bias detection in polish news portals. *Annals of Computer Science and Information Systems*, 15:21–24.

Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L. Alfonso Ureña-López. 2021. A survey on bias in deep nlp. *Applied Sciences*, 11(7).

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin

City University and Association for Computational Linguistics.

Halyna Padalko, Vasyl Chomko, and Dmytro Chumachenko. 2024. A novel approach to fake news classification using lstm-based deep learning models. *Frontiers in Big Data*, 6.

Manan Sharma, Krishanu Kashyap, Kushagra Gupta, and Shailender Kumar. 2023. Advancing political bias detection: A novel high-accuracy model. In *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, pages 347–352.

Timo Spinde, Lada Rudnitckaia, Kanishka Sinha, Felix Hamborg, Bela Gipp, and Karsten Donnay. 2021. Mbic–a media bias annotation dataset including annotator characteristics. *arXiv preprint arXiv:2105.11910*.

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