Explaining and Countering Media Bias: Toward Interpretable and Argumentative News Analysis

Sandesh Ghimire
sghimire26@amherst.edu

Shivangi Mittal
shivangishai@umass.edu

1 Introduction

Political polarization has amplified the need for automatic tools that highlight media bias and help readers contextualize news coverage. Bias in news media subtly shapes political opinion by framing events, omitting perspectives, or emphasizing selective facts.

Prior work on political bias detection has achieved high accuracy on classifying articles as left-,center-, or right-leaning. This project aims to extend political bias detection with two interpretive and interactive capabilities:

1. Explaining the model's decision, and highlighting the linguistic and semantic cues resposible for perceived bias
2. Generating counter arguments by articulating opposing viewpoint to encourage balanced understanding and critical reading.

By combining these two capabilities, we aim to connect bias detection and reasoning. The resulting product could support the general public, journalists, and educators in identifying and reflecting on ideological framing in political discourse.

2 Related work

Our project lies at the intersection of three growing areas in NLP: political bias detection, explainable reasoning, and counter-argument generation.

Early research on bias detection focused mainly on classifying news articles rather than explaining the underlying reasoning. Baly et al. (2020) developed a multi-task BERT model that predicts the political direction of an article along with its factual reliability. Their approach showed that contextual embeddings capture ideological cues effectively.

More recent work has emphasized transparency and interpretability in misinformation detection. Li et al. (2024) developed an explainable fake news detection system built on LLMs, introducing a "defense among competing wisdom" mechanism that forces the model to reason from multiple perspectives before making a prediction. This idea aligns closely with our goal of making bias detection more interpretable. In a complementary direction, Pan et al. (2023) introduced program-guided fact check, breaking down complex claims into smaller, verifiable steps. We plan to adapt this form of structured reasoning to decompose biased claims and generate counter-arguments grounded in logic and evidence.

The second part of our project builds on work in argument mining and text generation. Kawarada et al. (2025) surveyed how LLMs can identify argumentative components such as claims, premises, and reasoning chains from unstructured text. Their findings suggest that LLMs are capable not only of detecting arguments but also of understanding their internal logic. Expanding on this, Lin et al. (2023) developed a model for the generation of sentences-level counter-arguments, designed to target and refute weak premises. This area of research demonstrates that LLMs can reason about opposition, a key element in constructing meaningful counter-arguments to biased narratives.

Together, these studies highlight significant progress in explainability and argument generation, but remain separate efforts. Our work aims to bring them together: a unified system that explains why a model detects bias and offers a well-reasoned counter-narrative to challenge the article's framing.

3 Your approach

There are two main phases to our project based on what we are trying to achieve: (a) Bias Detection and Explanation, and (b) Counter-Argument Generation.

3.1 Phase 1: Bias Detection and Explanation

In this phase, we will identify and also explain biased language in a piece of text. With the help of the AllSides 21K dataset, we will fine-tune transformer-based models like BERT and RoBERTa. This will help with the classification of the political leaning of the text, which could be left, center, or right.

To achieve the task of interpretability, we plan to make use of rationale spans. These are phrases that mainly help the model's classification. In other words, these are the words or phrases that influence the decision of the model. For this, attention weights and gradient-based attribution are used. We plan to extract and visualize these phrases with the help of these attention weights and gradient-based attribution, like Integrated Gradients, LIME, or SHAP.

The rationales that our model would generate would then be compared with biases that are human-annotated, like emotionally charged words, so that the alignment with the human judgment can be evaluated.

In order to assess performance, we will be calculating the classification accuracy, precision, recall, and F1-score of the model. If the alignment is strong and the classification performance is high, it will indicate successful bias detection and interpretation.

The biased segments identified in this phase will serve as direct input to the second phase. This will allow phase 2 to focus on constructing counter-arguments for the same biased content.

3.2 Phase 2: Counter-Argument Generation

Once we have identified biased segments, we will use LLMs like `Llama-2-Chat` to generate counter-arguments that have an alternative viewpoint.

For instance, if an article portrays government spending as unnecessary, our model will generate a counter-argument with an alternative viewpoint, such that it frames it in a way that termed as an investment in valid cause like infrastructure or social welfare.

To show the effectiveness of our proposed model, we will use baseline LLMs like GPT-3.5 or Llama-2-Chat, which would be prompted without a bias rationale input to generate counter-arguments. We will then compare the argument generated by these baseline LLMs against those produced by our bias-aware model that makes use of rationale spans and ideological labels from Phase 1.

We plan to use metrics like BLUE, BERTScore, and factual consistency via FEVER/Natural language Inference scoring for the evaluation of both sets of output. The other metrics would be human judgment. This would help in evaluating how relevant, coherent, and factually correct the generated output is.

As our model knows where the bias is and also what type of bias it is, it can make responses that are more accurate and relevant than a normal LLMs that just try to reply without that context.

The motivation behind the combination of phase 1 and phase 2 is to create an interpretable and interactive system that not only detects ideological framing but also helps in critical engagement with news content.

3.3 Workflow

The diagram showing all the stages: (see Figure 1)

3.4 Baselines

We will compare our approach against two baselines:

- **Simple Baseline:** Logistic Regression using TF–IDF features.

- **Standard Baseline:** Fine-tuned BERT model, but without any of the additional features like explaining the bias or generating counter-arguments.

3.5 Schedule

- **Weeks 1–2:** Data gathering and preprocessing: gather and clean AllSides and other datasets; prepare tuples → (claim, argument, counter-argument).

- **Weeks 3–4:** Bias detection modeling: fine-tune `BERT/RoBERTa`, extract and visualize attention-based rationales.

- **Weeks 5–6:** Counter-argument generation: fine-tune `Flan-T5/Llama-2-Chat`

- **Week 7:** Integration and evaluation: integrate modules, evaluate using quantitative and qualitative metrics.

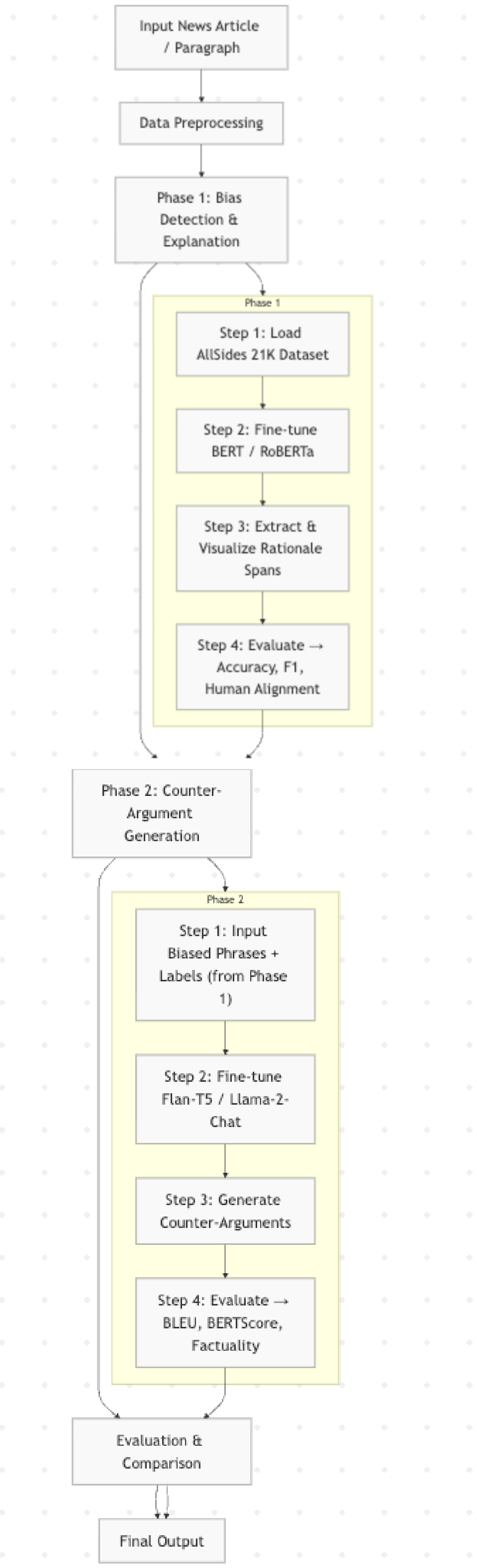
Input News Article / Paragraph
Data Preprocessing
Phase 1: Bias Detection & Explanation
Phase 1
Step 1: Load AllSides 21K Dataset
Step 2: Fine-tune BERT / RoBERTa
Step 3: Extract & Visualize Rationale Spans
Step 4: Evaluate → Accuracy, F1, Human Alignment
Phase 2: Counter-Argument Generation
Phase 2
Step 1: Input Biased Phrases + Labels (from Phase 1)
Step 2: Fine-tune Flan-T5 / Llama-2-Chat
Step 3: Generate Counter-Arguments
Step 4: Evaluate → BLEU, BERTScore, Factuality
Evaluation & Comparison
Final Output

Figure 1: System workflow showing input news article leading to bias classification, rationale highlighting, and counter-argument generation.

- **Week 8:** Final report for submission and presentation: documentation and summary of results and findings.

4 Data

For the bias detection and explanation phase, we will use the **AllSides 21K News Corpus** (Baly et al., 2020), a large dataset consisting over twenty-one thousand news articles labeled as *left*, *center*, or *right*-leaning. This corpus ensures consistency with prior experiments and provides a strong baseline for explainable bias modeling.

For counter-argument generation, we will explore several datasets designed for argumentation and reasoning:

- **IBM Debater Dataset**: annotated claims and counter-arguments across multiple controversial topics, suitable for training rebuttal-style models.

- **ArguAna / ArgPairs**: pairs of arguments and counter-arguments that can be used for stance detection and rebuttal construction.

- **Perspectrum**: a dataset linking claims with both supporting and opposing perspectives.

- Additionally, we may scrape user-generated debates from platforms such as **Reddit** to mine natural counter-arguments for open-domain generalization.

All datasets will be preprocessed into standardized tuples of the form `(claim, argument, counter_argument)` or `(article, bias_label, rationale)` for supervised fine-tuning and prompt-based generation. During training, textual preprocessing will include tokenization, lowercasing, and removal of boilerplate news content to ensure model robustness across sources.

5 Tools

We plan to use Python and Google Colab and also make use of GPU support so that we can fine tune our model.

5.1 Libraries and Frameworks

- **Transformers (Hugging Face):** so that we can fine tune BERT, RoBERTa, and Flan-T5 models

- **PyTorch:** to train our model and also for backpropagation visualization.
- **scikit-learn:** for baseline models like Logistic Regression. It will also help with evaluation metrics
- **LIME / SHAP:** to generate heatmaps. This helps in interpretability of indicators that help us in determining the bias
- **NLTK:** for tokenization, lemmatization, and text preprocessing.
- **Pandas & NumPy:** for data manipulation
- **Matplotlib:** for visualizing → attention maps, rationales. This will also be useful in comparing results

5.2 Environment and Infrastructure

- **Google Colab:** We will use Google Colab to run our code/experiment.
- **GitHub:** will be used to collaborate and for version control of the code.
- **Paid APIs:** As of now we are not planning to use any paid API.
- **Datasets:** The datasets that we plan to use are available publicly.

6 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes. When we write something in Overleaf, the AI underlines some words/ phrases, and when we click on it, it substitutes them with better phrasing.
  - We have used AI to help with an error in the code for workflow generation in mermaid.

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - ”Please let me know what could lead to following error in mermaid code Error: Error: Parse error on line 8: ...subgraph PHASE1[” ”] blank subgraph Expecting ’SEMI’, ’NEWLINE’, ’EOF’, got ’SPACE’ ”
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - The substitution of phrases made sentences clearer and concise.
  - The AI gave correct response for the error in the mermaid code we were able resolve the error

References

Baly, R., Karadzhov, G., Saleh, A., Glass, J., and Nakov, P. (2020). We can detect your bias: Predicting political ideology of news articles. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5007–5018.

Kawarada, D., Shindo, H., and Matsumoto, Y. (2025). Llms for argument mining: Detection, extraction, and structure generation. *arXiv preprint arXiv:2505.22956*.

Li, J., Wang, X., Zhang, H., and Xu, C. (2024). Explainable fake news detection with large language models via defense among competing wisdom. *arXiv preprint arXiv:2405.03371*.

Lin, Y.-L., Ma, W.-Y., and Chang, C.-H. (2023). Toward sentence-level counter-argument generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1556–1572.

Pan, L., Chen, J., Chen, W., Ren, X., and Schütze, H. (2023). Program-guided fact-checking: Interpretable verification through decomposition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.