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

Sandesh Ghimire
sghimire26@amherst.edu

Shivangi Mittal
shivangishai@umass.edu

1 Report Overview

This report presents the motivation, design, experiments, and open challenges of our final project on bias detection and stance-controlled counter-argument generation. We summarize the problem setting, describe the system we built (and how it evolved from our initial plan), and evaluate our models using both quantitative metrics and qualitative examples from real outputs.

All code, training notebooks, and instructions to reproduce our experiments are available in our public repository.[1] In particular, the repo includes the training pipeline, key hyperparameters, and scripts/notebooks used to generate the results reported in this document.

We follow the ACL-style formatting guidelines provided by the course template, adapting section structure where needed to match our project workflow. Our core contributions are: (1) a high-accuracy bias detection model with interpretable rationale extraction, and (2) a stance-conditioned counter-argument generator trained via teacher–student distillation. The results highlight both the effectiveness of supervised bias detection and the remaining difficulty of producing reliable, politically shifted counter-narratives.

2 Problem Statement

Political polarization has intensified public concern about how news media frames political events and, in turn, shapes public perception. Bias in news reporting is often subtle: it can emerge through word choices, selective emphasis, omission of alternative perspectives, or the use of emotionally loaded language that nudges readers toward a particular interpretation. While prior work on political bias detection has achieved strong performance in classifying articles as left-, center-, or right-leaning, most systems stop at producing a label. In practice, a label alone is not enough for critical reading: readers (and downstream users such as journalists and educators) need to understand *why* an article is perceived as biased and what an alternative framing might look like.

This project addresses the gap between accurate bias *classification* and meaningful *interpretation*. Specifically, we aim to build an interpretable and interactive system that takes a news article as input and produces:

1. **A predicted political bias label** from {left, center, right}.
2. **An explanation of the prediction** by identifying the linguistic and semantic cues most responsible for the model's decision (e.g., salient words/phrases, framing devices, or other indicators of ideological slant).
3. **A counter-framed rewrite** (or counter-argument) that expresses the *opposite* ideological perspective while remaining civil and coherent, enabling side-by-side comparison of how framing can change interpretation of the same event.

The first output provides a clear and automated assessment of leaning. The second output adds transparency by surfacing the evidence the system relied upon, helping users evaluate whether the prediction is justified or driven by superficial correlations. The third output goes beyond explanation by actively illustrating ideological framing: by generating an opposing viewpoint (or a rewritten version from an alternative stance), the system encourages balanced understanding and helps readers reflect on how political language, emphasis, and narrative structure influence perception.

Overall, our goal is to connect bias detection with reasoning and constructive re-framing. A

[1] Code repository: github.com/Sandesh816/models

system with these capabilities could support the general public in critical news consumption, assist journalists in auditing framing choices, and provide educators with tools to demonstrate how ideological perspectives can shape political discourse.

3 What we Proposed vs. What we Accomplished

Overall, **Phase 1 (bias detection)** followed the proposal closely and achieved strong classification performance, while **Phase 2 (counter-argument generation)** required pragmatic changes due to training instability (e.g., NaN losses) and the lack of reliable gold counter-argument references.

- **Data & preprocessing (modified).** Phase 1 used AllSides 21K. For Phase 2, instead of an external counter-argument dataset, we built a prompt-based dataset conditioned on Phase 1 outputs (predicted bias + extracted rationale phrases).
- **Bias detection models (accomplished).** Implemented TF–IDF + Logistic Regression and a transformer baseline; then fine-tuned a transformer for left/center/right detection with strong macro-F1 (Section 8).
- **Rationales (changed approach).** Rather than IG/LIME/SHAP/attention (unstable/inconsistent), we used **class-conditioned TF–IDF** to extract salient, interpretable rationale phrases.
- **Rationale validation (partial).** We provide qualitative inspection and error analysis with representative examples, but did not run a full human-alignment study due to time/annotation constraints.
- **Counter-argument generation (partial, modified).** We used a **teacher–student** setup with Flan-T5 (teacher pseudo-targets; student fine-tuning), but student training could be unstable; the final system emphasizes prompt-driven generation and teacher targets as the most reliable reference.
- **Evaluation & error analysis (modified, accomplished).** Without gold references, BLEU/ROUGE were not meaningful; we instead used a **stance-flip evaluation** by reclassifying generated text with the Phase 1 model, and report failure modes (no flip, short/degenerate outputs, repetition) in relation to rationale quality and training stability.

4 Related Work

Political bias classification has been studied in several forms, ranging from surface-level lexical models to contextual encoders that capture ideological patterns. Early work relied on lexicons, sentiment cues, or simple supervised models trained on news sources. With the rise of pretrained transformers, BERT-based models have become the standard for bias prediction, achieving strong performance on datasets such as AllSides (Baly et al., 2020). These models demonstrate that political leaning correlates with consistent linguistic patterns, including framing verbs, named entities, and evaluative descriptions.

Interpretability methods are increasingly common in politically sensitive NLP tasks. Attribution techniques such as Integrated Gradients (Sundararajan et al., 2017), LIME, and SHAP have been applied to classification models in order to highlight influential text regions. In misinformation research, structured reasoning systems have been proposed that encourage models to consider multiple perspectives before reaching a judgment, as in the multi-view reasoning framework of (Li et al., 2024). Other work explores decomposing claims into smaller factual units for verification, such as program-guided fact-checking (Pan et al., 2023).

A second line of research relevant to our work concerns arguments and counter-arguments. Surveys of argument mining with large language models show that pretrained models can identify claims, premises, and relationships at scale (Kawarada et al., 2025). Several studies investigate generating counter-arguments, often conditioned on specific claims or stance labels. For instance, (Lin et al., 2023) propose a system that generates sentence-level rebuttals by targeting weak premises. Recent work in stance-controlled generation extends beyond isolated claims to rewriting full passages, though reliably controlling ideological framing remains an open challenge.
Our project builds on both families of work. From the bias detection literature, we adopt supervised transformers as classification engines and apply

Split	#Docs	Left	Center	Right
Train	15,227	7,192	2,977	5,058
Val	3,264	1,542	638	1,084
Test	3,263	1,541	638	1,084
Total	21,754	10,275	4,253	7,226

Table 1: Dataset size, stratified split sizes, and label distribution.

interpretable feature selection to explain predictions. From argument generation research, we adopt the idea of counter-narratives but extend it by tying generation directly to classifier-identified rationales. Unlike prior work that treats these tasks separately, our system integrates detection, explanation, and rewriting in a unified pipeline.

5 Dataset

We used the **AllSides 21K News Corpus** (Baly et al., 2020), containing 21,754 news articles labeled as *left*, *center*, or *right*. This dataset supports both stages of our pipeline: (i) bias detection with rationale extraction, and (ii) counter-argument generation conditioned on the detected bias and supporting evidence.

Basic statistics. The label distribution is imbalanced: left (10,275), center (4,253), and right (7,226). We use a stratified split of 70/15/15 into train/validation/test (15,227 / 3,264 / 3,263). Table 1 summarizes the corpus and split sizes.

Why this is challenging? The task is challenging due to (1) class imbalance, (2) substantial topical diversity across articles, and (3) the presence of strong lexical/source cues that can act as shortcuts for bias prediction. Moreover, Phase 2 must generate coherent counter-arguments using *predicted* bias labels and automatically extracted rationales, which introduces error propagation from Phase 1.

5.1 Data Preprocessing

We removed non-linguistic artifacts (e.g., HTML remnants, malformed characters, duplicate whitespace) and dropped the non-informative index column. For Phase 1, we lowercased and concatenated fields (title, heading, source, tags, and article text) into a single input string, then tokenized using a BERT tokenizer with a fixed maximum length (250 tokens) for consistent batching. For Phase 2, each example is transformed into a structured prompt containing: the article text, the detected bias label, a target stance (opposite of detected bias; center→balanced), and extracted rationale phrases.

5.2 Data Annotation

Bias labels are provided by the dataset creators; we did not collect additional human annotations. For Phase 2, we primarily rely on automatic evaluation (Section 8) to measure whether generated counter-arguments move the predicted stance in the intended direction. This enables scalable evaluation, but does not directly measure human-perceived quality (e.g., factuality, persuasiveness, tone).

6 Baselines

We separate (i) Phase 1 **classification baselines** from (ii) Phase 2 **generation baselines + ablations**. In Phase 1, we use two standard, widely-adopted baselines that capture complementary modeling assumptions (lexical vs. sequential). We additionally trained a fine-tuned BERT classifier as a strong reference point, but we defer detailed discussion of its performance to the (Section 8) to keep this section focused on the baselines used for comparison.

All models use the same stratified train/validation/test split (70/15/15) as described in Section 5, and all hyperparameters were tuned only on the validation set.

6.1 Phase 1: Bias Detection Baselines

TF–IDF + Logistic Regression. A linear classifier trained on TF–IDF features (uni/bi-grams). This baseline is strong, fast, and interpretable (via feature weights), providing a lexical reference point for what cues drive predictions.

Bidirectional LSTM Classifier. A neural sequence model trained over token embeddings. This baseline tests whether a non-transformer neural architecture can capture longer-range contextual cues beyond lexical features while remaining simpler and cheaper than transformer-based approaches.

Note on transformer reference. We also fine-tuned a pretrained BERT model for Phase 1 and observed improved performance relative to the above baselines. Since BERT is not used as a "baseline" model family in our comparisons, we report and discuss its results in Section 8.

Reporting. We report accuracy and macro-F1 on the test split in Table 2. For the best-performing

Model	Acc.	Macro-F1	Recall
TF–IDF + LogReg	0.91	0.90	0.89
BiLSTM	0.98	0.98	0.89
BERT (fine-tuned)	0.99	0.99	0.99

Table 2: Phase 1 models on bias classification. We report baseline models, plus the fine-tuned BERT model.

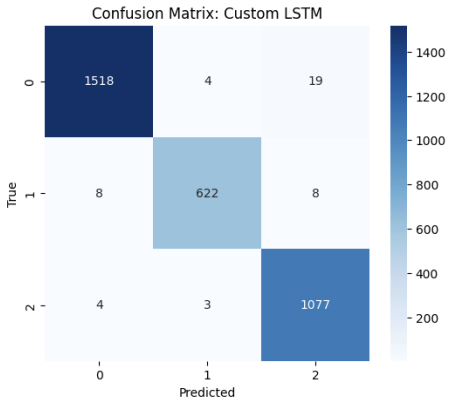

Figure 1: Confusion matrix for the Phase 1 LSTM. This diagnostic figure highlights which bias classes are most frequently confused.

Phase 1 model reported in Results, we additionally include a confusion matrix to show which labels are most frequently confused (e.g., center vs. left/right).

6.2 Phase 2: Counter-Argument Generation Baselines and Ablations

Phase 2 evaluates whether generated counter-arguments *flip stance* relative to the detected bias. We use the **flip rate** metric: after generating a counter-argument, we re-classify it with the same bias detector used for Phase 1 and count the fraction whose predicted stance matches the intended *opposite* stance.

Teacher–student setup. We use a larger FLAN–T5–XL model as a **teacher** to generate counter-arguments and a smaller FLAN–T5–base model as a **student** for distillation/fine-tuning. This design keeps inference efficient at test time while leveraging the teacher for higher-quality supervision.

Baselines (prompt-only generation). Our primary Phase 2 baseline is a **naive, generic counter-argument prompt** that does not condition on the Phase 1 predicted bias label or extracted rationales. This isolates how much stance flipping can be achieved from instruction-following alone.

Ablations (what changed). Beyond the naive baseline, we treat successive prompt and decoding modifications as **ablations** rather than separate baselines, because they share the same underlying generation objective and differ only in conditioning/decoding/selection. Table 4 summarizes the incremental effects of: (i) adding the predicted bias label, (ii) adding rationale phrases, (iii) tightening stance constraints in the prompt, and (iv) decoding and selection strategies (best-of-N, filtering, diversity schedules).

Interpreting the "strict prompt" drop. The non-monotonic flip rate (e.g., the strict prompt dropping to 26%) is informative: it suggests that *over-constraining* the instruction can reduce compliance or yield degenerate outputs, motivating the later best-of-N and filtering steps. Presenting this as an ablation keeps the narrative coherent.

7 Approach

Our system follows a two-phase pipeline designed to first identify ideological bias in news articles and then generate stance-controlled counter-arguments.

In **Phase 1**, we train a bias detection model based on a fine-tuned BERT classifier to categorize news articles into *left*, *center*, or *right* ideological classes. We evaluate multiple baselines, including TF–IDF with logistic regression and a BiLSTM model, and select BERT as the final Phase 1 model due to its superior performance across precision, recall, and F1-score metrics. In addition to producing a predicted bias label, the Phase 1 model supports interpretability by enabling the extraction of salient tokens associated with the predicted class. Using class-conditioned TF–IDF statistics, we identify high-weight terms that are strongly correlated with each ideological category. These extracted tokens are treated as *rationales*, representing bias-indicative cues present in the article.

The extracted rationales serve two purposes. First, they provide insight into the model's decision-making process by highlighting ideologically salient language. Second, they are incorporated into the downstream generation pipeline to better ground the counter-arguments in the original article content and its identified bias signals.

In **Phase 2**, we address the task of counter-argument generation using a teacher–student framework. A large FLAN-T5 model is used

Table 3: Phase 2 baselines. Flip rate is measured by re-classifying generated counter-arguments with a bias detector.

Generation Setup	Flip Rate	Notes
FLAN–T5 (base)	0.102	no bias label, no rationales, based only on prompt
Fine-tuned FLAN–T5	0.631	automatic eval via bias detector

as a teacher to generate high-quality counter-arguments conditioned on the article text, the predicted bias label from Phase 1, the target opposing stance, and the extracted rationales. Prompt engineering is used to explicitly specify the desired ideological stance and to encourage evidence-based reasoning rather than superficial stylistic opposition.

The teacher-generated counter-arguments are then used to fine-tune a smaller FLAN-T5-Base student model through knowledge distillation. This setup allows the student model to learn stance-aware argumentative patterns while maintaining lower computational cost at inference time. During generation, we apply controlled decoding strategies, including beam search and best-of-N sampling, along with quality filtering heuristics to remove overly short, repetitive, or weak outputs.

Finally, we evaluate the generated counter-arguments using a *stance-flip* metric. The Phase 1 BERT classifier is reused to assess whether a generated counter-argument successfully flips the predicted ideological stance relative to the original article. This automated evaluation enables large-scale measurement of stance control effectiveness and allows us to iteratively refine prompts, decoding parameters, and filtering strategies. Through this pipeline, the student model is trained to produce counter-arguments that challenge the original stance in a coherent and ideologically meaningful manner, rather than relying on generic or neutral language.

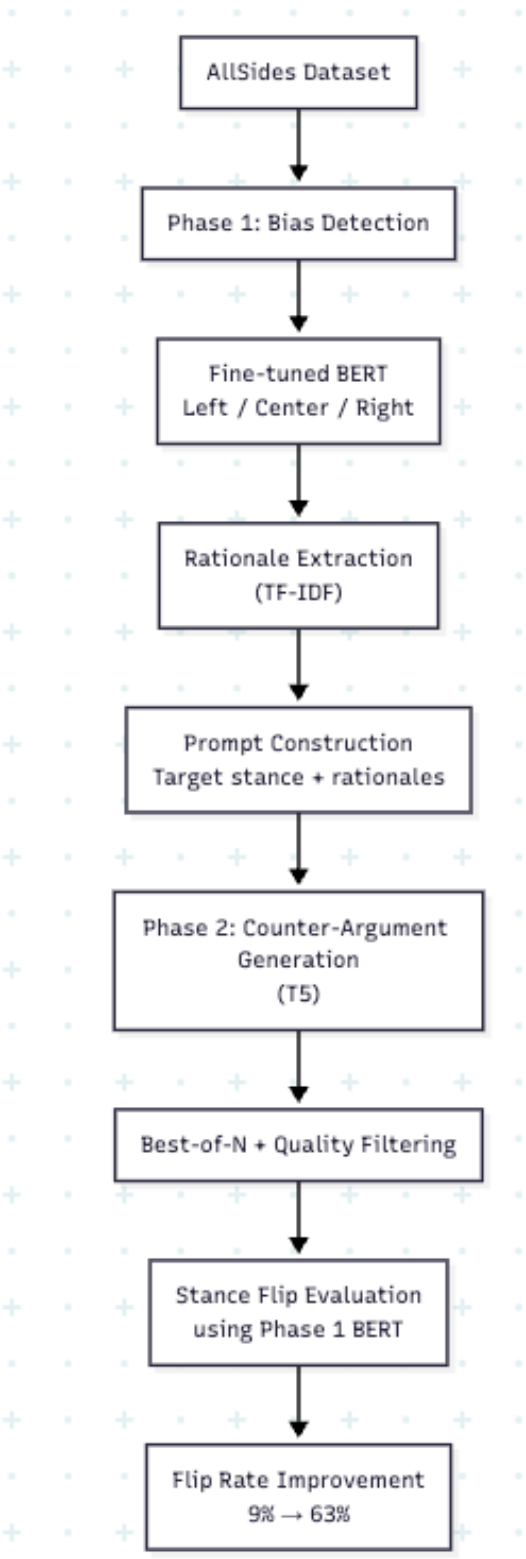

Figure 2: A flowchart summarizing our approach for the two-stage pipeline.

8 Results

8.1 Phase 1: Bias Detection Performance:

We used AllSides dataset to evaluate the performance of our bias detection models. After eval-

uating baselines like TF–IDF with Logistic Regression and a BiLSTM classifier, the fine-tuned BERT model seems to achieve the strongest performance across all metrics. Hence we selected this as our phase 1 model.

We see in Figure 3 that the classification report shows that the BERT model has consistently high precision, recall, and F1 scores across all three classes → left, center, and right. The overall in this case is 0.9905, and the macro-averaged F1 score is 0.99. This illustrates that although there is class imbalance,the model was able to generalize well.

We also see near-perfect separation between ideological classes in the confusion matrix (Figure 4). It also is very apparent that most of the error comes from 'center' labels which is expected because the centrist new content are somewhat ambiguous in nature. It can also be seen that confusion between the left and right classes is extremely rare, hence our model is suitable for this task.

We also see in the Receiver Operating Characteristic (ROC) curves (Figure 5) that there is strong discriminative capability, with an AUC of 1.00 for all three classes, which means that the model is able to maintain high true positive rates while minimizing false positives.

Finally, we observe from the distribution of maximum softmax probabilities (Figure 6) that the classifier is highly confident while making the predictions for the majority of examples. We also see that most samples concentrated near a probability of 1.0

```
----------Classification Report for BERT----------
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1541
           1       1.00      0.98      0.99       638
           2       0.98      0.99      0.99      1084

    accuracy                           0.99      3263
   macro avg       0.99      0.99      0.99      3263
weighted avg       0.99      0.99      0.99      3263


Accuracy: 0.9905
```

Figure 3: Classification report for the fine-tuned BERT bias detection model, showing precision, recall, and F1 scores across the left, center, and right ideological classes.

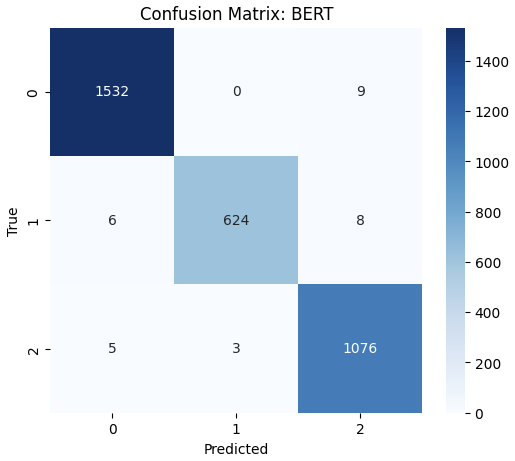

Figure 4: Confusion matrix illustrating near-perfect separation between ideological classes, with most misclassifications occurring in the center category.

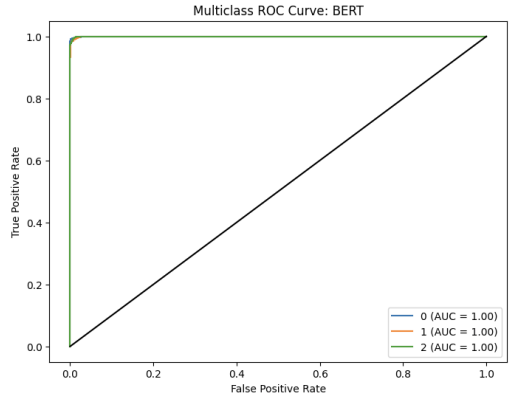

Figure 5: Receiver Operating Characteristic (ROC) curves for the bias detection model, demonstrating strong discriminative capability with an AUC of 1.00 for all classes.

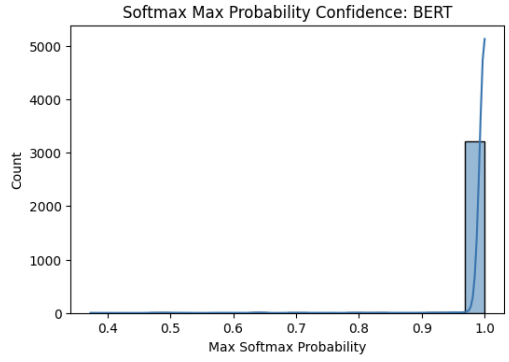

Figure 6: Distribution of maximum softmax probabilities, indicating high confidence predictions for the majority of samples.

8.2 Phase 2: Counter-Argument Generation and Stance Flip Evaluation

The evaluation of phase 2 involves the model's ability to generate counter-arguments that have an opposite perspective. We have used a stance-flip metric for this because we don't have any gold-standard counter-arguments for this purpose. We consider that the generated argument is successful if it is classified by the Phase 1 bias detector as belonging to the intended target stance.

We have also performed the evaluation only on articles with clear *left* or *right* ground-truth labels, excluding centrist articles to avoid ambiguity. Using a teacher–student framework with strict prompt conditioning, controlled decoding, and a best-of-N sampling strategy, the final system achieves a flip rate of approximately 63%, a substantial improvement over the 9% flip rate of the baseline model. As it can be seen from the table that Flip-rate improves steadily as explicit stance conditioning, decoding control, quality filtering, and reranking are introduced, with best-of-N sampling yielding the largest gains. Even if it's not perfect, the achieved flip rate demonstrates both the promise and the difficulty of stance-controlled generation, especially in politically nuanced domains

Table 4: Incremental Improvement in Phase 2 Flip Rate

Phase 2 Iteration and What changed	**Flip rate**
Baseline: Naive prompting without bias conditioning	∼10%
Bias-conditioned prompt: Added predicted bias label to the prompt	∼21%
Bias + rationales: Included extracted biased phrases as rationales	∼37%
Strict prompt: Enforced explicit stance adoption and disagreement	26%
Decoding constraints: Length control, repetition penalties, no-repeat n-grams	37%
Best-of-N sampling (N=5): Multiple generations with stance verification	∼50%
Quality filtering: Removed short and repetitive generations	∼57%
Decoding diversity: Temperature/top-p schedule across attempts	∼60%
Best-of-N sampling (N=7): Increased candidate pool for reranking	**63%**

9 Error Analysis

Although there was significant improvement in stance-controlled generation, several consistent failure modes remain. These are as follows →

Neutral or hedging language : Some counter-arguments avoid strong ideological commitment by adopting cautious phrasing like "some may argue...". While such outputs may appear valid to humans, they often fail to trigger a stance flip under automatic evaluation. This behavior reflects the inherent difficulty of enforcing strong stance control without encouraging overly extreme or repetitive language, particularly when the model is instructed to remain civil and evidence-based. This behavior reflects the inherent difficulty of enforcing strong stance control without encouraging overly extreme or repetitive language, particularly when the model is instructed to remain civil and evidence-based.

Error propagation from Phase 1: Because Phase 2 relies on predicted bias labels and extracted rationales, when there are errors in misclassifications in Phase 1, it can cause the generator to target a stance that is not correct, resulting in counter-arguments that are very similar to the original.This dependency highlights a key limitation of modular pipelines: while Phase 1 achieves high overall accuracy, even rare misclassifications can disproportionately affect downstream generation quality.

Topic drift and generic political statements: In these cases, the generated text expresses the target ideology but drifts to a topic that has nothing to do with the original text, reducing its effectiveness as a counter-argument. This limitation becomes worse when extracted rationales are broad or abstract rather than tightly connected to concrete claims.

Evaluation Limitations of the Stance-Flip Metric: Finally, some apparent errors stem from limitations of the automatic stance-flip evalua-

tion itself. Short but ideologically clear counter-arguments are occasionally misclassified by the bias detector, particularly when the generated text lacks strong lexical cues commonly associated with partisan framing. Conversely, some longer outputs that repeat stereotypical partisan language may achieve a flip despite limited argumentative depth

Overall, remaining errors can come from (i) insufficient ideological commitment, (ii) upstream bias misclassification, (iii) weak topical grounding, and (iv) limitations of automatic evaluation. Addressing these issues likely requires tighter coupling between rationale extraction and generation. It would also require improved discourse-level grounding, and complementary human evaluation.

10 Contributions of Group Members

Shivangi Mittal:

Conducted quantitative evaluation and visualization (classification reports, confusion matrices, ROC curves) for phase 1. Designed and implemented the rationale extraction component using class-conditioned TF–IDF features to identify salient bias cues, and integrated these rationales into the Phase 2 generation pipeline. Refined and incorporated prompt engineering, controlled decoding, best-of-N sampling, quality filtering in phase 2 of the system. Performed ablation studies, error analysis, and worked on the Approach, Results, Error Analysis, and Conclusion sections of the report.

Sandesh Ghimire:

Implemented major components across both phases of the project. In Phase 1, handled dataset pre-processing and exploratory analysis (including label inspection and stratified splits), implemented the TF–IDF + logistic regression baseline and a Bidirectional-LSTM model, and fine-tuned a BERT-based classifier. Led Phase 1 evaluation by producing the graphs and quantitative results reported in Section 8. In Phase 2, initialized and implemented the counter-argument generation pipeline , set up the teacher–student distillation framework, experimented with prompt-engineering strategies, and implemented the Phase 2 evaluation metric (stance flip rate). Drafted and refined the Problem Statement, Dataset, Baselines, What We Proposed vs. What We Accomplished, and Conclusion sections of this report.

11 Conclusion

This project explores a structured approach to countering political bias in news articles by combining bias detection, rationale extraction, and controlled generation. Our results suggest that explicitly modeling bias signals and leveraging teacher–student distillation improves the ability of language models to generate meaningful counter-arguments.

While our evaluation relies on automatic metrics, the observed improvements motivate future work involving human judgments, richer discourse-level evaluation, and tighter integration between detection and generation. Overall, this work demonstrates the promise of interpretable and modular NLP pipelines for addressing complex socio-political language tasks.

12 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.
- We have also used AI to help with code to generate tables in latex

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

- The AI helped with syntax on how to generate tables in latex

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*.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.