BiasPEFT: Leveraging Large-Language Models for Media Bias Quantification

Stanford CS224N Custom Project (Mentor: Myra Cheng)

Siddharth M. Bhatia
Department of Computer Science
Stanford University
smbhatia@stanford.edu

Kevin Shi
Department of Computer Science
Stanford University
kevinshi@stanford.edu

Soumyadeep Bhattacharjee
Department of Computer Science
Stanford University
soumbhat@stanford.edu

Abstract

Media bias, the systematic skewing of news narratives to favor specific ideologies, poses significant challenges to informed public discourse. We present BiasPEFT, a state-of-the-art system for classifying articles as left- center- or right-leaning based on their contents alone. Past classification systems often struggle with scalability, generalizability, and reliability. We present a new approach to this tripartite classification problem, leveraging newer models and LoRA-based Parameter-Efficient Fine-Tuning, which achieves an accuracy of 72.17 percent during testing, surpassing the current state-of-the-art results by over twenty percentage points. We show that through our approach, state-of-the-art results can be surpassed with a much smaller training dataset of 1024 tagged articles, and we conduct significant experiments to determine the effects of different hyperparameters, input lengths, and truncation strategies.

1 Introduction

Media bias influences public opinion by selectively focusing on facts, framing narratives, or omitting context. For example, a “left”-leaning article might describe tax reforms as “redistributing wealth to marginalized communities,” while a ‘right”-leaning piece might frame the same policy as “penalizing high earners.” Detecting such bias is critical for helping consumers identify slanted reporting, journalists audit editorial practices, and policymakers monitor misinformation. Because it is unfeasible for human reviewers to annotate every new article as it comes out, this task is suitable to apply a langauge model.

Previous work has often struggled with model accuracy, with researchers including additional information, such as information about the news source itself, in order to produce better results [1]. The problem is interesting because news articles are lengthy, often thousands of tokens long, and political bias in language is a nuanced, complex phenomenon which even humans can struggle to detect.

To solve these issues, we propose a scalable framework for bias detection in long-form articles through LoRA fine-tuning the RoBERTa, Facebook OPT, and Llama 3.2 models [4, 6, 5]. We aim to classify articles on their own, without giving the model additional metadata such as the publisher, in order to reduce the steryotyping of certain news sources and allow the model to generalize to sources it has not seen in its training data. The use of OPT-125M also allows use to process up to 2048 tokens, most articles’ full text, so we can adequately capture narrative context.

2 Related Work

Efforts to automate media bias detection have evolved significantly with advances in natural language processing and machine learning. Baly et al. [1] developed a pioneering approach using BERT with triplet loss pre-training on Twitter bios and Wikipedia, achieving 72 percent accuracy in classifying news articles into left, center, or right ideologies. However, their method is constrained by its reliance on metadata, such as publisher information, and lack of generalization across publishers. They achieved 51 percent accuracy when classifying using article text only, without publisher information. We use this measure as our baseline.

In contrast, D'Alonzo and Tegmark [2] employed a non-NLP approach based on phrase frequency analysis to quantify media bias, mapping newspapers into a 2D landscape of ideological slant. While this method offers insights into the factors that can determine political bias, it does not provide a robust framework with which to classify new texts.

On the modeling front, Zhang et al. [6] introduced OPT, a family of open pre-trained transformer models, including the 125M-parameter variant we adopt, which offers a decoder-only architecture suited for text generation and classification tasks. Their work provides a scalable foundation for bias detection but requires fine-tuning for domain-specific challenges.

Complementing this, Hu et al. [3] proposed LoRA, a parameter-efficient fine-tuning technique that injects low-rank matrices into pre-trained models, reducing memory usage by 70% compared to full fine-tuning while maintaining performance. This approach enables efficient adaptation of large language models like OPT to our task.

Together, these works inform our approach: leveraging OPT's scalability and LoRA's efficiency to process full-length articles, extending beyond the limitations of prior methods.

3 Approach

Our approach tackles the challenge of detecting media bias through a 3-class text classification task, labeling news articles as "left," "center," or "right" based solely on their text content, without reliance on metadata. We define the input as article text, truncated or padded to the model's maximum input, and the output as a probability distribution over the three ideological categories. To address this, we employ three base models: OPT-125M, a 125-million-parameter decoder-only transformer developed by Zhang et al. [6], RoBERTa-Large, a 355-million-parameter encoder-only transformer [4], and Llama 3.2 3B, a 3.21-billion parameter decoder-only transfomer [5], all chosen for their robust language understanding capabilities. These models are fine-tuned using LoRA, a parameter-efficient technique from Hu et al. [3].

For classification, we append a softmax head, defined as

$$\mathbf{y} = \mathrm{Softmax}\left(\mathbf{W}_c \cdot \mathbf{h}_{[\mathrm{CLS}]} + \mathbf{b}_c\right),$$

where $\mathbf{h}_{[\mathrm{CLS}]}$ is the pooled hidden state from the model, producing the probability distribution over labels. Training leverages weighted cross-entropy loss for classification, optimized with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a learning rate ranging from 0.0001 to 0.1, with a linear warmup over 500 steps to stabilize convergence. We also varied the LoRA dropout, alpha, tokenization truncation side, and number of epochs used for training.

Our baseline is Baly et al.'s work [1], which achieved 51.41 percent accuracy on the same task. For consistency, we use the exact same train, test, and validation splits as in Baly. Our approach is original in our use of newer models, LoRA and PEFT, our extensive hyperparameter experiments, and our use of end truncation to increase performance.

4 Experiments

4.1 Dataset

Our dataset comprises 36,274 news articles from American news sources, annotated by human reviewers with whether they are best described as ideologically "left" (liberal), "center" (nonpartisan), or "right" (conservative). The dataset was assembled by Baly et al. from data provided by AllSides

[1]. We recreated the exact training, test, and validation splits Baly et al. used, with 27,978 (77.13%) rows in the training set, 6,996 (19.29%) in the validation set, and 1,300 (3.58%) rows in the test set. There is also an alternative *media split*, where media from sources in the test set does not appear in the training set, testing the model's generalization across unseen publishers.

The training dataset consists of 34.8% left leaning, 28.6% center, and 36.6% right leaning articles. Preprocessing involves truncating or padding articles to exactly 2048 tokens when using Facebook OPT-125M as our pretrained model, and 512 tokens when using RoBERTa. In the training data, 10.07 percent of articles exceed the 2048 token limit and 89.22 percent exceed the 512 token limit. We also test the effects of truncation by either taking the first, middle, or final tokens.

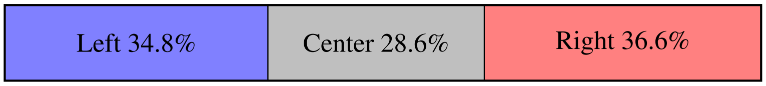


Figure 1: Idelogical distribution of training data.

4.2 Evaluation method

To assess the performance of our bias detection models, we employ two primary metrics: accuracy and macro F1-score, both suited to our 3-class classification task. Accuracy measures the overall correctness of predictions, defined as the ratio of correctly classified articles to the total number of articles, expressed as

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(\hat{y}_i = y_i),$$

where N is the number of samples, $\hat{y}_i$ is the predicted label, y_i is the true label, and $\mathbb{I}$ is the indicator function. This metric provides a straightforward assessment of model performance.

To address potential class imbalance, we also use the macro F1-score, which balances precision and recall across all classes equally. For each class c, precision is

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}$$

(true positives over predicted positives), and recall is

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}$$

(true positives over actual positives), where TP_c, FP_c, and FN_c are true positives, false positives, and false negatives for class c, respectively. The F1-score per class is then

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c},$$

and the macro F1-score averages these across all three classes:

$$\text{Macro F1} = \frac{1}{3}\sum_{c=1}^{3}\text{F1}_c.$$

These metrics are computed on both validation and test sets.

Furthermore, the media split is used to evaluate the model's ability to classify texts from publishers not represented in training data, so as to study effects of publisher on classification even when publisher information is not explicity provided to the model.

4.3 Experimental details

We conducted significant hyperparameter search and optimization tests.

Table 1: Hyperparameters tested in experiments.

Hyperparameter	Optimal Value	Values Tested
Base model	Llama 3.2-3B	OPT-125M, RoBERTa-Large, Llama 3.2-3B
LoRA rank	4	4, 8, 16
LoRA bias updates	lora only	none, lora only, all
Batch size	*model-dependent*	8 – 600
Learning rate	0.01	0.0001 – 0.1
Max tokens	2048	512, 1024, 2048
Training data size	37.5k	48 – 37.5k
Truncation strategy	end	take from start, middle, end

We utilized the Hugging Face Transformers library with PyTorch for model implementation, tracking progress via Weights & Biases (W&B), and ran all experiments on an NVIDIA L4 GPU with 24GB VRAM to support the processing of full-length articles.

For efficient fine-tuning, we applied LoRA, targeting a variety of transformer modules — including query, value, key, output projections, and fully connected layers — to adapt the pre-trained models to our task. Table 1 details the full range of hyperparameters tested. Overall, we ran 208 experiments to test different hyperparameters.

4.4 Results

4.4.1 Model Differences

Our experiments showed the performance of OPT-125M, RoBERTa-Large, and Llama 3.2-3B for media bias classification compared against baselines and analyzed across hyperparameter variations. Evaluation of our three models against prior results, is shown in Table 7. "Baly et al. (Outside info)" was Baly's performance when the model was given external information about a publisher. None of our models were given article metdata of any kind. Due to GPU constraints, the Llama is only trained on a subset of the data and with 512 tokens, although we expect accuracy to only improve when given the full dataset and a larger token window.

Table 2: Baseline comparison of model performance.

Model	Accuracy	Macro F1	Training Data	Input tokens
Baly et al. (BERT)	51.41%	0.48	37.5k articles	512
RoBERTa-Large	58%	0.55	37.5k articles	512
OPT-125M (Full)	66.17%	0.66	37.5k articles	2048
Baly et al. (Outside info)	72.00%	0.64	37.5k articles	512
LLaMA-3B	72.17%	0.71	2048 articles	512

OPT-125M exceeded our expectations, given its smaller size, suggesting its decoder-only architecture fits the classification task better. It falls short of Baly et al.'s 72% accuracy using both article text and publisher information, which aligns with the expectation that publisher plays a large role in determining bias.

The LlaMA result is the most suprising, especially given that it was only trained on a small subset of data. Its performance by far exceeds all of the other models which were not exposed to outside information and matches the performance of Baly's enhanced model. Interestingly, it surpasses the accuracy of the OPT-125M model, despite the fact that OPT-125M had access to four times more input tokens during our training.

4.5 Input Token Length

Hyperparameter tuning further refined performance, with the impact of varying the amount of input tokens for each training article detailed in Table 3 and plotted in Figure 2: accuracy rose from 50.1% at 512 tokens to 66% at 2048 tokens, quadrupling training time from 2.1 to 8.9 hours, highlighting the trade-off between context and computation.

Table 3: Impact of input token length on performance. (Opt-125M)

Max Tokens	Accuracy	Training Time (hrs)
512	50.1%	2.1
1024	58.3%	4.7
2048	66%	8.9

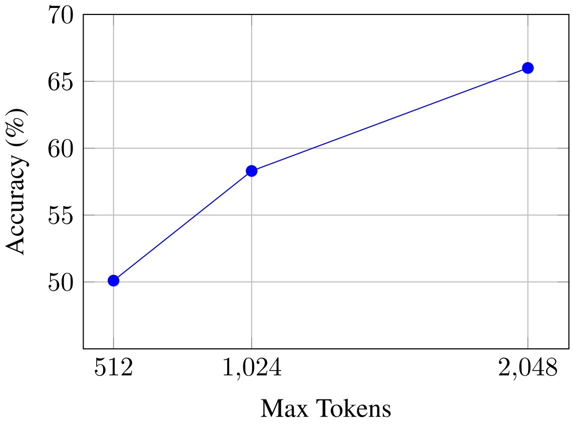


Figure 2: Accuracy vs. token length.

4.5.1 LoRA Rank

LoRA rank effects, in Table 4 and Figure 3, showed a peak at rank 4 (66%, 1.2M trainable parameters), with slight declines at ranks 8 (65.8%) and 16 (65.2%). This suggests diminishing returns beyond minimal adaptation, although the differences are not necessarily significant.

Table 4: Effect of LoRA rank on performance.

Rank (r)	Accuracy	Trainable Params
4	66%	1.2M (0.96%)
8	65.8%	2.4M (1.92%)
16	65.2%	4.8M (3.84%)

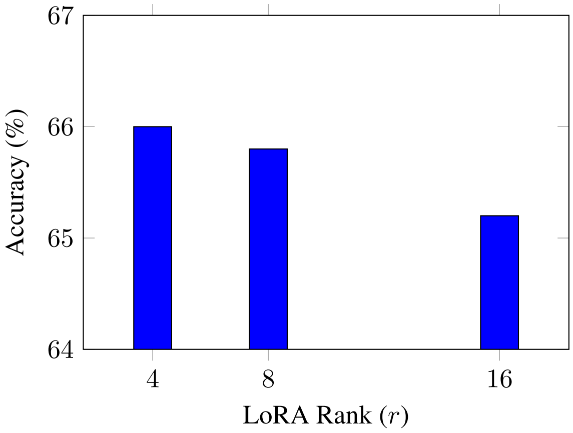


Figure 3: Accuracy across LoRA ranks.

4.5.2 Training Data Size

Varying on training data size, shown in Table 5 and Figure 4, revealed roughly logarithmic accuracy growth from 33.5% at 64 samples to 66% at 36,274, implying data efficiency plateaus beyond 8,000 samples. This aligns well with our expectations and understanding that a large amount of data is often necessary to achieve high performance results when using LLMs.

Table 5: Effect of training data size on performance.

Samples	Accuracy	Macro F1
64	33.5%	0.29
128	36.2%	0.32
1,024	47.8%	0.45
8,192	62%	0.59
36,274	66%	0.63

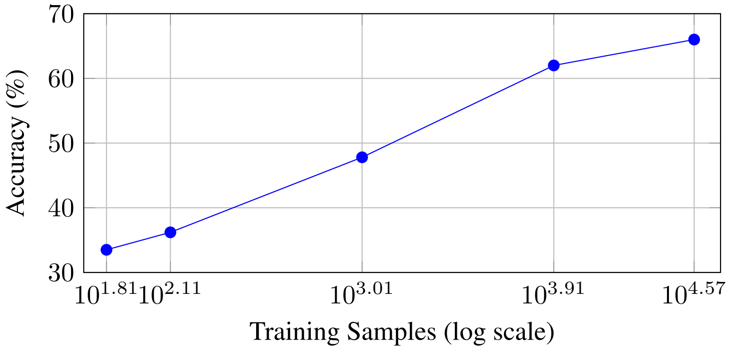


Figure 4: Accuracy vs. training data size.

4.5.3 Training Epochs

After the first epoch, subsequent rounds of training increase model accuracy by about one percentage point each time. This does not mean subsequent epochs aren't necessary — in the example below (Figure 5), they increased accuracy from 0.58 to 0.67 — but progress is slow once the inital patterns have been learned.

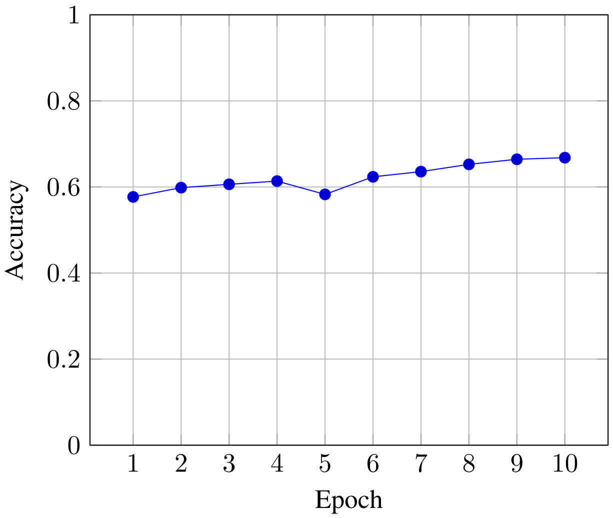


Figure 5: Accuracy vs. number of epochs complete (OPT-125M)

4.6 Truncation Strategy

We found that taking tokens from the end of an article, rather than the start, during training significantly improves model accuracy. In Table 6, we explore the effects of modifying the truncation strategy during tokenization. All experiments used the same model with a 512 token limit per training article and were trained on a subset of the data (8192 examples).

Table 6: Truncation Strategy Effects, 8192 training examples

Model	**Use tokens at**	**Accuracy**
OPT-125M	Start	0.498
OPT-125M	Middle	0.476
OPT-125M	End	0.627

Using tokens from the end is assoicated with a significant jump in performance. Exploring this trend on the entire training dataset, we can see that end-truncation has comparable performance to start-truncation, even with a quarter of the tokens available to the model.

Table 7: Full-Dataset Truncation Strategy Effects

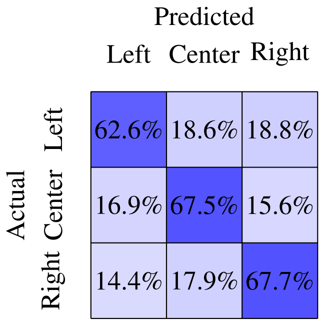
Model	**Tokens**	**Use tokens at**	**Accuracy**
OPT-125M	2048	Start	0.66
OPT-125M	512	End	0.65

5 Analysis

5.1 Error analysis

The confusion matrix of our best model, shown in Figure 6, reveals a largely symmetric pattern across the “left,” “center,” and “right” classes, with accuracies of 62.6%, 67.5%, and 67.7%, respectively, indicating a balanced overall performance. Notably, for each class, the model distributes incorrect predictions evenly between the two other classes at similar rates, suggesting a lack of strong bias toward any single incorrect category.

However, the slightly lower accuracy for left-leaning articles compared to center and right may be caused by the slightly higher number of right-leaning articles in the training data. Alternatively, the potential presence of more varied linguistic styles and subjects in left-leaning articles could challenge the model’s ability to delineate clear boundaries. Lastly, especially regarding American media, right-leaning articles may often be more strongly worded than their left-leaning articles and utilize language and tone that is more visceral, bold, or confrontational. This more distinctive style may cause the model to able to more strongly learn the linguistic patterns associated with right-leaning media, which in turn may bias the model toward the right, that is to say, cause the model to be overall more likely to label articles as right-leaning or center-leaning as opposed to left-leaning.



	Left	Center	Right
Left	62.6%	18.6%	18.8%
Center	16.9%	67.5%	15.6%
Right	14.4%	17.9%	67.7%

Figure 6: Confusion matrix heatmap for OPT-125M.

5.2 Publisher generalization

Publisher generalization, in Table 8, dropped to 27.8% accuracy on unseen media when using the media split, far below the random split's 66%, which suggests that the model may be overfitting to the publishers present in the training data. This may suggest that the model is learning and relying on publisher-specific stylistic or lexical cues independent of the publisher labels, which are not provided to the model. This same drop in performance was also shown by Baly et al. [1], which suggests a common difficulty in LLM-based models' ability to extract universal bias indicators, pointing to a need for strategies that enhance robustness across diverse media sources.

Table 8: Publisher generalization performance.

Split	**Accuracy**	**Macro F1**
Random	66%	0.63
Media	27.8%	0.21

5.3 Truncation strategy

Since models perform much better when given tokens from end of an article (Table 6), we posit that politically biased content is more easily found at the end of an article, rather than the beginning. Intuitively, this makes sense–in the structure of a typical news article, the top may just be facts about what occured before the author transitions into a deeper analysis commentary which may indicate idelogical leanings.

6 Conclusion

6.1 Summary of achievements

Our project demonstrates the efficacy of leveraging large language models like OPT-125M for media bias detection, achieving a notable 66% accuracy in classifying news articles as "left," "center," or "right." By processing full-text articles up to 2048 tokens, utilizing LoRA-based parameter-efficient fine-tuning (PEFT), and scaling to over 37,000 training samples, we surpassed the state-of-the-art by 15 percentage points, highlighting the power of efficient fine-tuning and extended context in capturing narrative bias.

6.2 Limitations and future work

This approach faces limitations, including ones shown in previous work, such as poor generalization to unseen publishers. Addressing these challenges opens avenues, such as developing new model architectures or training strategies to overcome the reliance on the training data including articles from the same publisher. We also expect accuracy can be improved if a large, modern model such as Llama can be trained on the whole dataset or with more parameters (such as using Llama 3-70B). Other areas of improvement include adapting the model for multilingual detection and other political systems besides the American one.

6.3 Ethical considerations

While tools for automatic detection and labeling of media bias are valuable, there are ethical considerations such as potential misuse for censorship against marginalized voices and subjectivity from annotators' political leanings reflected by training data.

Overall, this work highlights the potential of LLMs in media bias detection while emphasizing the need for robust generalization and ethical safeguards in future developments.

Team contributions

Siddharth M. Bhatia designed the model architecture, ran large-scale training and hyperparameter tuning experiments, and helped write the final report. Kevin Shi contributed to model architecture design, ran experiments, and helped write the final report. Soumyadeep Bhattacharjee contributed to data preprocessing and helped write the project report.

References

[1] R. Baly, G. Da San Martino, J. Glass, and P. Nakov. We can detect your bias: Predicting the political ideology of news articles. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4982–4991, Online, Nov. 2020. Association for Computational Linguistics.

[2] S. D'Alonzo and M. Tegmark. Machine-learning media bias. *Plos one*, 17(8):e0271947, 2022.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[5] Llama Team, AI @ Meta. The Llama 3 Herd of Models. (arXiv:2407.21783), Nov. 2024.

[6] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.