Text Mining Final Assignment
Group 12
Biased Sentences In News

Manuel Pérez Belizón*
s4416880
Wan Rong Shen*
s2400804

March 27, 2025

1 Introduction

Biased news and media framing significantly undermine people's right to access accurate information, as misinformation can shape or distort public understanding and perspectives. This issue has become increasingly concerning in the digital age, where biased and misleading news content spreads rapidly through channels like news websites and social media. Many studies have applied Natural Language Processing (NLP) models, including transformer-based models like BERT and large language models (LLMs) like GPT-4o to identify biased news content by training on human-labeled data. However, such training data often include only a single rating per article, overlooking the possibility that annotators with differing levels of knowledge about the news events may provide varying labels.

The dataset created by Lim et al. (2020) [2] includes labels from annotators with both preknown and not preknown knowledge of news events, this paper explores whether including annotators' knowledge statuses into the training process can improve the performance of a few-shot BERT model and GPT-4o on biased news classification task. The classification task uses a four-point scale: Neutral, Slightly biased, Biased, and Very biased. Specifically, we investigate how integrating annotators' knowledge statuses affects the models' overall performance.

Our research questions are as follows:

1. How does including annotators' preknown and not preknown status impact the overall performance of BERT and GPT-4o models in biased news classification?

2. What strategies can be employed to aggregate labels from preknown and not preknown annotators of the same article during model fine-tuning?

2 Related Work

Biased news classification NLP research has been developed for a long time, many researches have applied NLP models using many different techniques and models for bias news detection.

This paper uses and compares the dataset introduced in the paper of Sora Lim, Adam Jatowt, Michael Farber and Masatoshi Yoshikawa [2] this

dataset introduces the preknowledge of the annotator as well as a more specific classification with a 4 point scaled that goes from Very biased to No biased.

It seems that the most recent papers in the field of biased news classification are using different transformer models like BERT, in the work of Shaina et al. (2024)[5] they used BERT to detect biases in text by predicting NER labels.

Very recent researches [3, 4] in this field suggest that pre-trained neural transformers like the GPT family can outperform other models.

3 Data

3.1 Train Dataset

The dataset from Lim et al. (2020) primarily used in this paper consists of 46 unique news articles, each annotated by multiple annotators, resulting in a total of 215 bias labels.

The news articles cover four news events: (1) the suicide of a state lawmaker accused of sexual assault, (2) debates over Facebook's handling of personal data and calls for regulation, (3) Donald Trump's criticism of NFL players protesting during the national anthem, and (4) U.S. Secretary of State Rex Tillerson's statement on being prepared to negotiate with North Korea without preconditions.

Each news event is represented by a similar number of articles in the dataset, and the distribution of annotator labels for each event is shown in Table 1. The labels could be divided into two annotator groups: `preknow=yes` and `preknow=no`, indicating whether the annotator had prior knowledge of the news event before providing bias labels.

Among all the 46 articles, 43 have labels from both `preknow=yes` and `preknow=no` annotators, and 3 articles only have labels from `preknow=no` annotators.

Annotator Knowledge Status	**News Event 1**	**News Event 2**	**News Event 3**	**News Event 4**	**Total**
`preknow=no`	40	36	42	38	**156**
`preknow=yes`	10	21	16	12	**59**
Total	**50**	**57**	**58**	**50**	**215**

Table 1: Number of labels of 4 Events for `preknow=yes` and `preknow=no`.

Lastly, the bias labels are categorized into four classes: Neutral, Slightly biased, Biased, and Very biased. Table 2 shows 91% of the labels fall under the categories of Neutral, Slightly biased, and Biased, with Biased (42.79%) and Slightly biased (33.49%) being the most common among them.

We observed that `preknow=yes` annotators assigned fewer "Biased" or "Very biased" labels than `preknow=no` annotators. The uneven distribution of ratings between the two groups of annotators suggests that prior knowledge led `preknow=yes` annotators to make more moderate judgments.

Annotator Knowledge Status	**Neutral**	**Slightly biased**	**Biased**	**Very biased**	**Total**
preknow_no	14	46	78	18	**156**
preknow_yes	18	26	14	1	**59**
Total	**32**	**72**	**92**	**19**	**215**

Table 2: Counts of Bias Labels across the annotator groups preknow_no and preknow_yes

3.2 Additional Train Dataset: BABE

The primary challenge with the given dataset is its small size, which has led to continuous fluctuations in the training loss during the training process. This raises our concerns that the model's performance may not be reliable or meaningful. To address this, we searched for additional news bias research datasets. However, most existing datasets classify news into only two broad categories, such as left vs. right or biased vs. non-biased, making them incomparable with the given dataset.

Therefore, we chose the BABE dataset proposed by Spinde et al. (2021) [6]. This dataset

contains approximately 3,700 sentences, with labels based on a majority vote of all annotators. The labels in this dataset are divided into three categories: Entirely factual, Somewhat factual but also opinionated, and Expresses writer's opinion. To ensure compatibility, we converted these three labels to match the categories of the first dataset, as shown in Table 3.

Final Label	**Count**
Biased (Expressed writers opinion)	858
Slightly biased (Somewhat factual)	1000
Neutral (Entirely factual)	1600
Total	**3458**

Table 3: Labels of additional human labeled bias news dataset BABE.

3.3 Data Preprocessing

For the first dataset the preprocessing that we had to do can be summarized in the following steps:

1. We calculate means and majorities for the different `preknow` labels having 4 different new fields:
    - Preknow yes mean
    - Preknow yes majority
    - Preknow no mean
    - Proknow no majority
2. We create the full text article combining the different sentences.
3. We combine all the unique articles using the selected label aggregation, this will be discussed in the next section.

The second dataset requires almost no preprocessing as the labels are already aggregated based on a majority vote approach.

4 Methods

In this section we will discuss the different ways we handle and aggregate the labels and the approaches followed to use the different models.

4.1 Label Handling and Aggregation

The paper suggests aggregating the labels from `preknow=yes` and `preknow=no` annotators to achieve a more balanced representation, this could help to mitigate the subjective biases of informed annotators (`preknow=yes`) and the limited context of uninformed annotators (`preknow=no`).

To handle labels in this paper and to assess the impact of including annotators' knowledge information on model performance, we tried four different approaches:

- **Prioritize preknow=yes**: If `preknow=yes` labels are available for an article then we use the mean of these labels and ignore the `preknow=no` labels. If `preknow=yes` labels are not available then we fall back to using the mean of the `preknow=no` labels.

  This approach prioritizes the labels from the informed annotators (`preknow=yes`) based on the observation in the paper [2] that their labels are generally more consistent and reliable.

  The result labels are shown in Table 4, where we can see that the `Very biased` label is lost after the aggregation.

Label (After Aggregation)	**Count**
Biased	10
Slightly biased	23
Neutral	13
Total	**46**

Table 4: Final labels after using Prioritize preknow=yes aggregation

- **Weighted preknow=no and preknow=yes (0.5)**: We assign equal weights (0.5) to the `preknow=no` and `preknow=yes` labels. As shown in Table 1, the ratio of `preknow=no` to `preknow=yes` is approximately 0.75:0.25, so this weight (0.5) still prioritizes `preknow=yes` while ensuring the contributions of `preknow=no` to the final labels.

Label (After Aggregation)	Count
Very biased	8
Biased	12
Slightly biased	22
Neutral	4
Total	**46**

Table 5: Final labels after using 0.5 weighting (0.5) for `preknow=yes` and `preknow=no` aggregation

- **Treat preknow=yes and preknow=no equally**: In this approach, we ignored the annotators' information and treated all labels equally. For the BERT model, all ratings were fed into the model, resulting in the same outcomes as shown in Table 2.

  For GPT-4o mini, directly using all news articles posed challenges due to the presence of sensitive words, leading to fine-tuning failures flagged by the moderation system.

  To address this issue, we used the most frequent label (majority) to determine the final label for each article, regardless of the annotators' preknown status. The resulting labels are shown in Table 6. We could see this method resulted in an increase in the number of articles labeled as "Biased" compared to the previous 2 methods.

- **No preknow + extra dataset**: In this approach we used the before mentioned [6] extra dataset as a way to train our models and then our original dataset to test our models.

Label (After Aggregation)	Count
Very biased	3
Biased	22
Slightly biased	18
Neutral	3
Total	**46**

Table 6: Final labels after using no or majority aggregation (ignore `preknow=yes` and `preknow=no`)

4.2 Model Training

After aggregating the labels of `preknow=yes` and `preknow=no` for each article, the dataset size decreased from 215 to only 46 unique labeled articles. It was thus not feasible to train a model from scratch on such a small dataset. Therefore, we employed pretrained foundation models, including BERT (Large, Cased) and GPT-4o mini, for the classification task. We compared the performance of the models before and after fine-tuning. The complete model settings are presented in Table 7.

Model	Fine-tuned	Train Data
GPT-4o mini	No	Default
GPT-4o mini	Yes	Included annotators' information
GPT-4o mini	Yes	Not included annotators' information
GPT-4o mini	Yes	Not included annotators' information + more data
BERT	No	Default
BERT	Yes	Included annotators' information
BERT	Yes	Not included annotators' information
BERT	Yes	Not included annotators' information + more data

Table 7: Overview of Models with Fine-tuning and Training Data Settings

4.3 BERT

The BERT model was built using Keras [1]. The exact architecture of the model is the following:

Layer	Params	Connected to
Input Layer	0	[]
Bert Preprocess	0	[Input Layer]
Best Encoder	28763649	[Bert Preprocess Layer]
Dropout	0	[Bert Encoder Layer]
Classifier head output	2052	[Dropout]

Table 8: BERT model architecture

The specific BERT model we are using is small_bert/bert_en_uncased_L-4_H-512_A-8, to prevent over-fitting we have a Dropout layer with a dropout ratio on 0.1 and we have a Dense layer as a classifier head to classify the four possible categories.

To train this model we used a 80-10-10 train,test and validation split, using Adam as the optimizer with a learning rate of 1e-5, sparse categorical loss as our loss function and we trained the model for 20 epochs with early stopping.

4.4 GPT-4o mini

The GPT-4o mini model was trained on a total of 82,761 tokens over 3 epochs with a batch size of 1. The learning rate multiplier was set to 1.8 for optimized training. For the model input, fine-tuning was performed using the required chat-completions training file in JSONL format. Each text entry in the file included structured prompts, such as:

> *"You are a helpful assistant that classifies the news article into one of the following classes based on its text content: ['Neutral', 'Slightly biased', 'Biased', 'Very biased']" and lists the class at the end of your response.*

and followed with the news text as user input. The completions represented the corresponding classification labels.

5 Results

In this section, we present the results from our various experiments and approaches.

5.1 BERT

The results obtained using the BERT model for different approaches are summarized in the table below:

Aggregation Method	**Accuracy**
Prioritize preknow=yes	0.60
Weighted preknow (0.5, 0.5)	0.59
Weighted preknow (0.8, 0.2)	0.61
No preknow	0.42
No preknow + extra dataset	0.33
Only with BABE dataset	**0.71**

Table 9: Results of BERT model using different training methods

5.2 GPT-4o-mini

The results for the GPT-4o-mini model are as follows:

Aggregation Method	**Accuracy**
Prioritize preknow=yes	0.4
Weighted preknow (0.5, 0.5)	0.2
No preknow	0.2
Untrained zero-shot	0.5
No preknow + extra dataset	**0.7**
Trained only with BABE dataset	**0.7**

Table 10: Results of GPT-4o-mini model using different training methods

6 Discussion

6.1 BERT

From the results shown in Table 9, we can see that the model performs better when the information provided by annotators with preknown knowledge is prioritized over that from annotators without preknown knowledge. This supports the findings in [2], which suggest that annotators with preknown information tend to produce more consistent and reliable labels.

However, one limitation of the original dataset is its small size, which likely constrained the model's ability to train effectively. This limitation is evident when comparing the performance using

only the extra dataset.

Furthermore, when the extra dataset was used as the training set and the original dataset as the test set, the model struggled. We believe that differences in bias types across the datasets impacted the model's ability to generalize effectively.

6.2 GPT-4o-mini

From Table 10, we observe a noticeable increase in accuracy as the training dataset size grows. Specifically, when the BABE dataset is included, the accuracy reaches 0.7, indicating that, for both the BERT and GPT-4o-mini models, dataset size has the most significant impact on model performance, especially when the training data is too small.

Regarding aggregation methods, when using the Prioritize preknow=yes approach, the GPT-4o-mini model achieved the highest accuracy of 0.4, outperforming other aggregation methods. This also confirms the findings in [2] that the annotators' preknown knowledge help producing more reliable classifications.

However, the Untrained zero-shot model still performed slightly better, achieving an accuracy of 0.5, suggesting that even without fine-tuning, the model was able to generate reasonable classifications.

Overall, the results highlight the critical role of dataset size in enhancing model performance, particularly when a larger datasets like BABE are included. Adding such datasets makes the training loss much more stable during the fine-tuning process and increases the robustness of models in biased news detection.

7 Conclusion

This paper explored the impact of annotators' prior knowledge on the performance of BERT and GPT-4o models in biased news classification tasks. Our findings suggest that incorporating annotators' preknown information generally improves model performance, which aligns with existing research indicating that more informed annotators produce more reliable labels. However, for large language models like GPT-4o-mini, the model itself was already capable of classifying news articles effectively without prior knowledge.

While the Prioritize preknow approach was beneficial, we observed that the primary factor impacting model performance was actually the size of the dataset. This was demonstrated when the BABE dataset was used for fine-tuning, which not only stabilized the training loss but also enhanced the model's robustness in detecting bias across news articles.

In summary, the inclusion of annotators' preknown status positively impacted model performance, but it was clear that dataset size had a more substantial effect, especially when fine-tuning with the external BABE dataset.

These findings emphasize the critical role of dataset augmentation and fine-tuning in biased news classification. Future work could explore the effects of different transformer models, additional datasets, and more sophisticated label aggregation methods to improve classification accuracy and enhance model generalization across various contexts.

8 Contributions of the team members

Section	**Manuel Pérez Belizón**
Introduction	Research problem definition, formulation of research questions
Related Work	Literature review, past work analysis
Data	Data collection, dataset description
Methods	Model training (BERT and GPT-4o), aggregation strategies
Results	Results presentation (BERT model)
Discussion	Discussion of BERT results
Conclusion	Drafted the conclusion based on findings

Table 11: Contributions of Manuel Pérez Belizón

Section	Wan Rong Shen
Introduction	Literature review, dataset explanation
Related Work	Literature review, model comparison
Data	Data preprocessing, dataset analysis
Methods	Fine-tuning, implementation of experiments
Results	Results presentation (GPT-4o model)
Discussion	Discussion of GPT-4o results
Conclusion	Provided feedback, final edits

Table 12: Contributions of Wan Rong Shen

References

[1] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[2] Sora Lim, Adam Jatowt, Michael Färber, and Masatoshi Yoshikawa. Annotating and analyzing biased sentences in news articles using crowdsourcing. In *Proceedings of the Twelveth International Conference on Language Resources and Evaluation, LREC 2020*, 2020.

[3] Terry Ruas Akiko Aizawa Bela Gipp Timo Spinde Martin Wessel, Tomáš Horych. Introducing mbib – the first media bias identification benchmark task and dataset collection. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

[4] Tim Menzner and Jochen L. Leidner. Experiments in news bias detection with pre-trained neural transformers. In *Advances in Information Retrieval*, 2024.

[5] Deepak John Reji Syed Raza Bashir Chen Ding Shaina Raza, Muskan Garg. Nbias a natural language processing framework for bias identification in text. In *Expert Systems with Applications, Volume 237, Part B*, 2024.

[6] Timo Spinde, Manuel Plank, Jan-David Krieger, Terry Ruas, Bela Gipp, and Akiko Aizawa. Neural media bias detection using distant supervision with babe - bias annotations by experts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, page 1166–1177. Association for Computational Linguistics, 2021.