Predicting Media Bias in Sentences

Sujen Kancherla
skancherla@ucsd.edu

1 Introduction

With an increasing presence online, most people are consuming a large amount of information through news and media, which may not be in an unbiased form. This is very harmful, especially in the context of politics and social issues. This paper presents an encoder using transformer architecture with BPE (Byte Pair Encoding) tokenization to classify text as bias or not. Using this architecture, the model should be able to pickup contextual information in the text that will lead to it being either neutral or biased of some sort.

- Collected and preprocessed dataset: DONE.
- Build and train a DAN neural network on the BABE dataset and examine its performance: DONE
- Make transformer-based model perform better than baseline model DONE.
- ~~Use pretrained BERT model on as encoder before classifying the bias or not and evaluate performance in comaprison to other models~~: NOT DONE: The training for the baseline and tranformer model was taking too long, and I did not have time left.

2 Related Work

2.1 LLM Annotations for Dataset Labeling

Horych et al. (Horych et al., 2024) explored the use of LLM's to annotate datasets. This piece is interesting because it speculates that LLM's may be able to synthetically annotate media data in most preprocessing stages of bias detection.

2.2 Transformer-Based Bias Detection

Menzner and Leidner (Menzner and Leidner, 2024) used a pretrained neural transformer models to detect bias in news articles. They showed that transformers are exceling at picking up the contextual information in order to analyze bias. The study also benchmarked the performance of these models against traditional methods, highlighting the transformers' ability to adapt to complex, high-dimensional text data.

2.3 Advancements in Content Analysis

Raza et al. (Raza et al., 2024) also used transformer based models for content analysis, specifically Bias detection. They also predicted the implict and explicit bias through the model, which may give it more understanding of bias if it learns where its coming from.

2.4 Comparative Studies on GPT Models

Wen and Younes (Wen and Younes, 2023) compared the performance of GPT-3.5 against fine-tuned language models in the task of bias detection. Their results showed that fine-tuned models often outperformed general LLMs due to their ability to focus on task nuances. This study underscores the importance of task-specific adaptation in improving bias detection.

2.5 Benchmarking Media Bias Detection

Wessel et al. (Wessel et al., 2023) introduced the MBIB dataset, the first benchmark collection designed specifically for media bias detection tasks. Their work provided a comprehensive evaluation framework, enabling the systematic comparison of various approaches. This dataset and benchmark methodology are particularly valuable for developing and evaluating our proposed model.

3 Your dataset

The most important rule of NLP: look at your data! Provide examples from your dataset, and clearly describe your task. Explain what properties of the data make your task challenging. Report the

source of the dataset, its basic statistics (e.g., size, number of words/sentences/documents) and some other statistics that are specifically relevant to your task. Show a couple of input / output pairs to make it clear what you're doing (but don't use up too much space in doing so!).

The dataset mainly used in the project is the BABE (Bias Annotations By Experts). The dataset consists of 2 subgroups, SG1 and SG2. They have 5 and 8 annotators, respectively. The final bias label is taken as an aggregate from the annotators where a split in votes is considered "No agreement." The task is to classify the text sentences as Biased or Not Biased. Here are the total numbers for the combined dataset:

- **Total examples:** 7,074
- **Number of "No agreement" examples:** 304
- **Number of Bias examples:** 3,574
- **Number of No Bias examples:** 3,196

Biased Sentence Example

Young female athletes' dreams and accomplishments have been dashed across the country when competing against biological males who "identify" as females.

Unbiased Sentence Example

Yet Biden finds himself in an increasingly competitive race with Sanders, the U.S. senator who came close to winning the 2016 Nevada caucus and finds support with some of the same voters.

3.1 Data preprocessing

The preprocessing steps were just combining the two subgroups into one big dataframe with all the examples. It was filtered to only have the Bias Label, Bias Words, and the text for each example. The only labeled kept were the Bias and No Bias labels, removing the no agreement examples. We hope the model would perform better with just having to do binary classification, instead of adding an uncertainty class. The Bias words are used in the final model to allow the tranformer to learn the specific bias words in context rather than just classifying the entire sequence.

3.2 Data annotation

If your project involves annotation, you may have started a pilot annotation experiment, annotating a few dozen or few hundred examples. What major issues have come up? What is your stance on the examples you annotated? If applicable, report any relevant observations or considerations regarding consistency in your annotations.

The BABE dataset was taken from a balanced content selection. The dataset was annotated by trained media bias research experts. Both of the subgroups in the dataset are annotated by multiple experts, and the final label of Bias or not is an aggregate of all the experts' labels. The other column used in the model is the bias words, this was also created using an aggregate of the bias words annotated by the experts.

4 Baselines

What are your baselines, Additionally, explain how each one works, and list the hyperparameters you used and how you tuned them. Describe your train/validation/test split. If you have tuned any hyperparameters on your test set, expect a major point deduction.

The baseline model is a **Deep Averaging Neural Network (DAN)** utilizing **Byte Pair Encoding (BPE)** as the tokenizer. The process is as follows:

1. **Tokenization**: Each input sentence is tokenized using BPE with a vocabulary size of $V = 100{,}277$ and a sequence length capped at $L = 32$.

2. **Embedding Averaging**: The embeddings ($d = 50$) of the tokens are averaged to produce a fixed-size representation of the input text.

3. **Network Architecture**:
   - **Input Layer**: The averaged embeddings serve as the input.
   - **Hidden Layers**: One fully connected linear layer, with $H = 128$ neurons, is applied with **ReLU activation**.
   - **Output Layer**: A softmax layer produces logits for the binary classification of **Biased** vs. **Non-Biased**.

Hyperparameter	Value
Vocabulary Size (V)	100,277
Sequence Length (L)	32
Embedding Dimension (d)	50
Hidden Layer Size (H)	128

Table 1: Baseline Model Hyperparameters

Model Hyperparameters

4.1 Data Split

The BABE dataset was split into 70 percent training, 15 percent validation, and 15 percent testing.

- **Training examples:** 4,739
- **Validation examples:** 1,015
- **Test examples:** 1,016

5 Your approach

The approach used in this project is to utilize a transformer based encoder model that classifies both a binary bias or not task, as well as biased tokens for each token in the text. This gives the model more understanding in the classification task to predict the actual bias words/tokens along with whether or not the entire sequence is biased.

5.1 Working Implementation

A fully working implementation of the approach was achieved. The transformer model used was fine-tuned using Hugging Face's pre-trained transformers as a base. The key components implemented include:

- **Binary Classification Head:** A feed-forward neural network attached to the encoder's output for sequence-level classification.
- **Token Classification Head:** A similar neural network that operates on token embeddings from the encoder for word-level bias detection.

The following files in the submitted code are associated with these models:

- `models/tranfromers.py`: Contains the architecture definitions for the binary and token classifier.
- `train.py`: Includes the training loop, data preprocessing, and optimization logic.
- `preprocess.py`: Includes data preprocessing.

5.2 Compute

The experiments were run on a local macbook pro, with an Intel i-9 processor.

5.3 Runtime

In total, **1 minute 43 seconds** were required to reach convergence across **10 epochs** on the baseline model. For the final model it took **14 minutes 33 seconds** to train over **20 epochs**.

5.4 Results

The results achieved by the model on the test set are as follows:

- **Binary Classification Accuracy:** The model achieved an accuracy of **84.32%**, outperforming the baseline model accuracy of **75%**.

These results demonstrate an improvement over baseline models, validating the effectiveness of the dual-task learning approach.

5.5 Other Details

Additional important details include:

- The model used `Adam` optimizer with a learning rate of **0.001**.
- The loss function was a weighted combination of cross-entropy losses for both tasks.
- Weight decaying was implemented to regularize overfitting.

5.6 Performance Over Epochs

Figure 1 shows the training, validation, and test accuracies over epochs, highlighting model convergence and generalization performance.

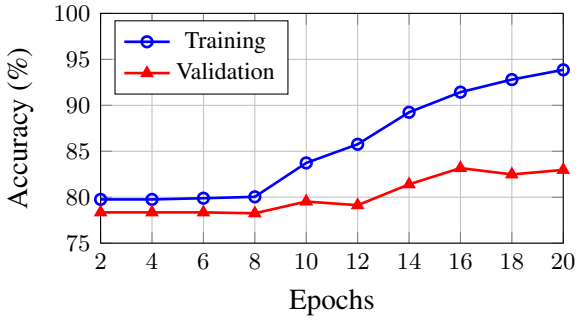

Figure 1: Performance of the model over epochs for training, validation.

6 Error analysis

6.1 Baseline Model

Misclassified Example 1

Sentence:
"The slogan of 'defund the police' has arisen in the wake of rioting and looting that hit cities around the country following the killing of George Floyd."

Predicted Label: Biased
True Label: Non-biased

Analysis: The model overemphasized terms like "defund," "rioting," and "looting," associating them with bias even though the overall tone is neutral.

Misclassified Example 2

Sentence:
"The Trump tax cuts also doubled the child tax credit and expanded its eligibility, which greatly helps the American working-class family."

Predicted Label: Non-biased
True Label: Biased

Analysis: The nerual network was not able to understand the context and classify correctly over a larger sequence. The second part of the sequence is where most of the bias come from, but it may only be picking up the semantics in the first part

6.2 Final Model

Misclassified Example

Sentence:
"The deputy commissioner of Kolkata Police eventually apologized for the bloodying of students - many of whom were women - and argued that tensions had risen too quickly."

Predicted Label: Biased
True Label: Non-Biased

Bias tokens Predicted: "eventually" and "argued"

Analysis: The final model is much better at picking up the nuanced biases than the baseline. In this example too, it recognizes "eventually" and "argued" as tokens that may indicated a biased sequence. But the biggest issue with the model is it sometimes focuses heavily on words instead of the overall meaning.

6.3 Model Differences

The baseline model is more of a direct approach with embedding the sequence and a neural network to classify the sequence. The errors arise in this approach with larger sequences, the average embedding will not be able to learn long range context over a large sequence. The final model, on the other hand, has its limitations with more focus on context and individual tokens. This may be slightly more accurate than the baseline, but it also tends to focus heavily on key words/tokens that are related to bias. This may not always be accurate when the sequence is nested with contradicting clauses or even referencing other text.

7 Conclusion

Bias detection in general as a task for humans or models, can be very subjective and highly dependent on data and the views represented in it. In the BABE dataset, used in this project, there are multiple annotators to try to avoid the human error in classifying bias. Even with multiple annotators, there may be amiguity and uncertainty with some text. The results from the experiments are as expected because the baseline model was a simple neural network. The transformer based model was able to outperform it because it not only learned contextual information in the attention blocks, but

it classified each token as bias or not. It had more annotations to learn from. In the future one approach that may enhance the architecture is to use a pre-trained encoder model, like BERT to generate the embeddings for the text and then send the output to a classifier for bias or not predictions.

8 Acknowledgements

Acknowledging the foundational contributions of Vaswani et al. (Vaswani et al., 2017) for the Transformer architecture, which serves as the backbone of modern deep learning in NLP, and Sennrich et al. (Sennrich et al., 2016) for Byte Pair Encoding (BPE), which enabled efficient tokenization for this project. Their pioneering work has been instrumental in advancing the field and directly influenced this implementation.

References

Horych, T., Mandl, C., Ruas, T., Greiner-Petter, A., Gipp, B., Aizawa, A., and Spinde, T. (2024). The promises and pitfalls of llm annotations in dataset labeling: a case study on media bias detection.

Menzner, T. and Leidner, J. L. (2024). Experiments in news bias detection with pre-trained neural transformers.

Raza, S., Bamgbose, O., Chatrath, V., Ghuge, S., Sidyakin, Y., and Muaad, A. Y. (2024). Unlocking bias detection: Leveraging transformer-based models for content analysis.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Wen, Z. and Younes, R. (2023). Chatgpt v.s. media bias: A comparative study of gpt-3.5 and fine-tuned language models. *Applied and Computational Engineering*, 21(1):249–257.

Wessel, M., Horych, T., Ruas, T., Aizawa, A., Gipp, B., and Spinde, T. (2023). 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*, SIGIR '23, page 2765–2774. ACM.