NEWS BIAS NEUTRALIZATION

PROJECT REPORT
FOR PHC-351

Nithish Ravikkumar
23120328

PROJECT OBJECTIVES

1

Detect Ideological Bias present in News and Media text reports using transformer-based classifier (BERT)

2

Rewrite subjectively biased text into a neutral point of view

3

Evaluate output neutrality using semantic similarity and bias-score metrics.

EXISTING WORK

Neural-Based Statement Classification for Biased Language (Hube & Fetahu, 2013)

identify the bias-inducing word in a sentence.

Limitation: Performance is similar to human test takers, and is focused on resolving epistemological bias.

Linguistic Models for Analyzing and Detecting Biased Language (Recasens et al., 2013)

Classifies a sentence or word as "biased" or "neutral" by identifying the bias-inducing word.

Limitiation: It does not resolve the bias.

Automatically Neutralizing Subjective Bias in Text (Pryzant et al., 2019)

Neutralize subjective bias in text using Transformer based model.

This paper has been used as a basis for this project, with significant use of the original model and dataset.

MOTIVATION

Subjective Bias

- Presenting opinions as facts, using loaded language, or biased framing—is a pervasive problem in news .
- Maintaining a Neutral Point of View to maintain trust and academic integrity of information sources.

Pryzant et al., 2019 requires high computational capacity for training of the model. This high computational barrier makes the model difficult to reproduce or adapt for most researchers.

The paper's automated metrics—**BLEU** and **Accuracy** (Exact Match)—are poor proxies for the actual goal of neutralization.

- The authors even admit this in Section 4.2 (Table 5), stating there is a "weak association between BLEU and human evaluation scores."
- My main contribution through this project is to improve on the evaluation metrics for measurement of bias correction.

MODEL ARCHITECTURE

- Pryzant et al. propose 2 different models in their paper - **Modular** and **Concurrent.**

- **Modular :** Two stage BERT+LSTM based model with a distinct Detector and editor stages, connected by a join embedding. The Editor is explicitly guided to "pay attention" to and rewrite the words the Detector flagged as biased, while leaving the rest of the sentence intact.

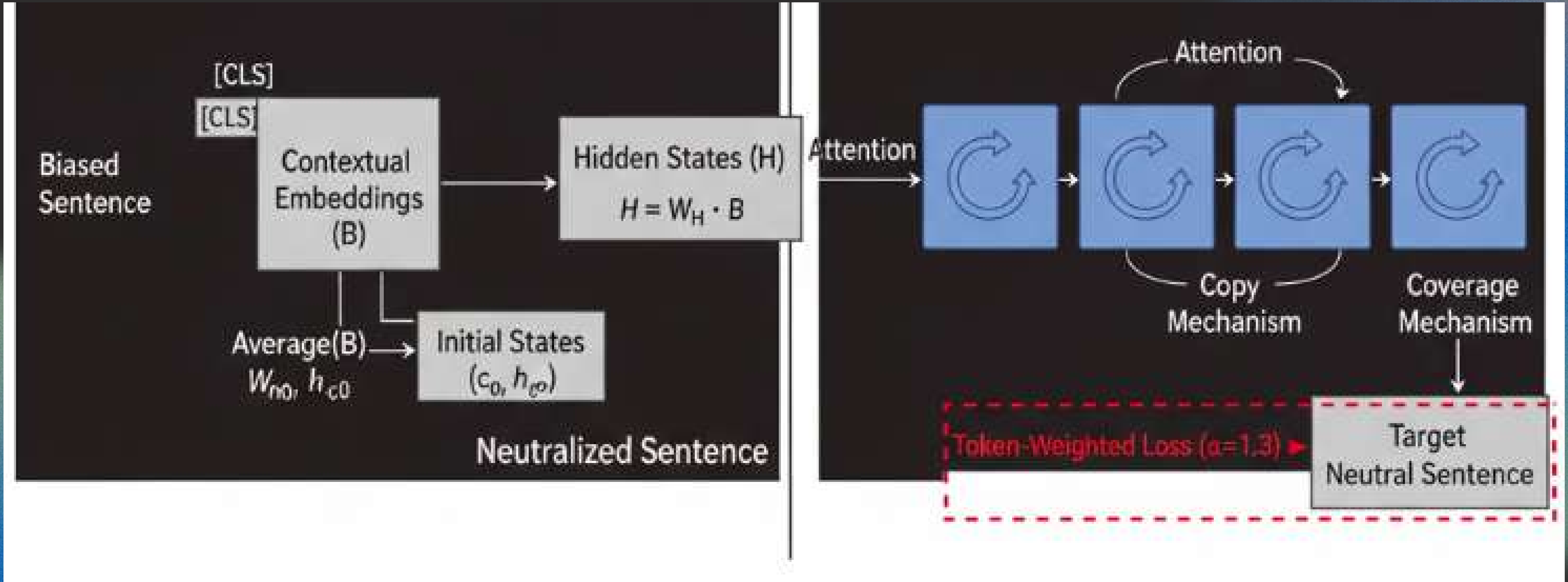
- **Concurrent :** treats neutralization as a standard sequence-to-sequence "translation" task, translating a biased sentence into a neutral one directly. (Single stage)

- In this project, I have proceeded with the Modular model.

Architecture:

- (Detector): BERT-based sequence tagger that analyzes the sentence and outputs a probability of bias for each individual word.
- (Editor): This is an attentional LSTM (sequence-to-sequence) model. It receives the original sentence's hidden states, but these states are modified by the Detector's output.

Encoder (BERT) block Decoder (LSTM+Attention) block



IMPLEMENTATION

Framework: Pytorch
Optimizer: Adam , Learning Rate: 5e-5 , Batch Size: 16, Epoch=10
Hardware: 2x NVIDIA T4 GPUs (Kaggle)

The code is divided into various files, each holding necessary functions-

Configuration (Config.py) : Hyperparameters- It sets crucial model and training parameters:

- BERT_MODEL_NAME: Uses 'bert-base-uncased' as the encoder.
- LSTM_HIDDEN_DIM: Sets the LSTM's hidden size to 768 to match BERT's output.
- EPOCHS, BATCH_SIZE, LEARNING_RATE: Standard training settings.
- LOSS_ALPHA: Sets the special token-weighted loss parameter to 1.3, which is a key detail from the paper.

Data Loading (Datafile.py): For Data generation, formatting and preparation for processing.

- This file reads the tab-separated value (TSV) files (like biased.word.train) using pandas.
- Contains Gemini API endpoint to generate Synthetic Biased-Neutral Datapoint pairs
- It tokenizes the source (biased) text for the BERT encoder.
- It tokenizes the target (neutral) text for the LSTM decoder.

Model :

- As discussed earlier, this contains the model definition and architecture used

Evaluation, Utility and metrics (eval.py and util.py): Evaluates the model on a dataset (validation or test set). It reports:

- **Loss:** The average token-weighted loss.
- **Accuracy:** The percentage of predicted tokens that exactly match the target.
- **BLEU Score:** A measure of sentence similarity. It uses the model.generate() method to get the model's full predicted sentences and compares them to the ground-truth neutral sentences.
- **Semantic Score:** To be discussed.
- **Aggregated Bias Score:** To be discussed.

Model Training (train.py):

- Setup: Loads the Config, BertTokenizer, and data loaders, builds the model
- Train Loop: Iterates for the specified number of EPOCHS, Trains the model on the train_loader, using the token_weighted_loss. After each epoch, runs calculate_metrics on the validation (dev_loader) set.
- Saves the model checkpoint (.pt file) if the validation loss improves.
- Runs calculate_metrics on the test_loader and prints the final Test Loss, Accuracy, and BLEU score.

EVALUATION METRICS

Original Evaluation Metrics (Pryzant et al., 2020)

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbb{I}(\text{pred}_i = \text{ref}_i)}{N}$$

BLEU

- N-gram overlap between machine translation output and reference translation
- Compute precision for n-grams of size 1 to 4
- Add brevity penalty (for too short translations)

$$\text{BLEU} = \min\left(1, \frac{\textit{output-length}}{\textit{reference-length}}\right) \left(\prod_{i=1}^{4} \textit{precision}_i\right)^{\frac{1}{4}}$$

Evaluation Gap

Problem 1: Exact Match is Too Strict

Problem 2: BLEU Penalizes Good Paraphrases

SOLUTIONS

SEMANTIC SIMILARITY

Measures meaning preservation between the original and neutralized text.

A high score indicates better semantic preservation.

This score is a weighted average of two methods to capture both deep meaning and key-term overlap.

$$\mathrm{Sim}_{\mathrm{SBERT}} = \frac{E_{\mathrm{orig}} \cdot E_{\mathrm{neut}}}{\|E_{\mathrm{orig}}\| \cdot \|E_{\mathrm{neut}}\|}$$

$$\mathrm{Sim}_{\mathrm{Jaccard}} = \frac{|W_{\mathrm{orig}} \cap W_{\mathrm{neut}}|}{|W_{\mathrm{orig}} \cup W_{\mathrm{neut}}|}$$

$$\mathrm{Sim}(S_{\mathrm{orig}}, S_{\mathrm{neut}}) = (0.8 \times \mathrm{Sim}_{\mathrm{SBERT}}) + (0.2 \times \mathrm{Sim}_{\mathrm{Jaccard}})$$

AGGREGATE BIAS SCORE

$$\text{Bias}(S) = \frac{P_{\text{detector}} + S_{\text{lexicon}} + (1 - P_{\text{neutral}})}{3}$$

A composite score from 0.0 (Neutral) to 1.0 (Biased), averaging three factors:

- The maximum bias probability (0.0–1.0) for any token, as reported by the fine- tuned Detector Module.
- A score (0.0–1.0) based on the frequency of words found in 14 known subjectivity/bias lexicons.
- The probability (0.0–1.0) that the text is “Neutral” from a separate, pre-trained ideological classifier.

PERFORMANCE, RESULTS & DISCUSSION

- Fine-tuned the **pre-trained Modular checkpoint** for 10 epochs on a hybrid (**WNC + LLM-generated**) dataset using Kaggle T4 GPUs.
-
- Beyond BLEU: The slightly lower BLEU score is expected, as the model learned to generate new, valid neutralizations.

Metric	Pryzant et al. (Modular)	Our Adapted Model
Hardware	NVIDIA TITAN X	Kaggle 2x T4
Training Strategy	(~10h)	Fine-tuning (~5.5h)
Exact Match (Accuracy)	75.49%	**57.2%**
BLEU Score	45.8	32.55
Semantic Similarity	(Not Measured)	**0.84 (High)**
Avg. Bias Score (Input)	(Not Measured)	0.78 (High)
Avg. Bias Score (Output)	(Not Measured)	**0.31 (Low)**

Proof of Neutralization: The high **Semantic Similarity (0.84)** and significant drop in **Bias Score (0.78 → 0.31)** demonstrate strong neutralization performance.

SCOPE FOR IMPROVEMENT

1. LSTM+Attention block can be replaced with a transformer decoder.

2. Domain Adaptation via Web Scraping.

3. Refine Synthetic Data

4. Integrate Fact-Checking

REFERENCES

- Pryzant, R., Martinez, R. D., Dass, N., Kurohashi, S., Jurafsky, D., & Yang, D. (2020). "Automatically Neutralizing Subjective Bias in Text." Proceedings of the AAAI Conference on Artificial Intelligence.
- Recasens et al. (2013). "Linguistic Models for Analyzing and Detecting Biased Language"
- Christoph Hube and Besnik Fetahu (2013). "Neural Based Statement Classification for Biased Language".
- Wiki Neutrality Corpus dataset: https://www.kaggle.com/datasets/chandiragunatilleke/wiki-neutrality-corpus
- News Media bias dataset (for reference, and synthetic data generation): https://huggingface.co/datasets/newsmediabias/news-bias-full-data

THANKS

PITCH DECK

Presented by

Claudia Alves