Detecting Bias and Generating Neutral Language Replacements with T5

W266 Final Project - Spring 2022
Lindsay Ng & Amanda Smith
{lindsay.ng, amandasmith}@berkeley.edu

Abstract

Subjective bias present in literature and media can intentionally or unintentionally presuppose truth or cast doubt through words. Such language can be nuanced and difficult to detect, making it potentially harmful by prejudicing viewpoints unbeknownst to the reader. We provide a solution for proofreading biased text in the form of a T5 encoder-decoder transformer model that detects biased words and generates a neutral language replacement. We experiment with training size, learning rate, number of epochs, and architecture variants to develop a high performance bias-neutralizing model. We find that the best performing model takes a large amount of training data as input and approaches state-of-the-art scores in Accuracy, BLEU, BLEURT, and BERTscore metrics.

Introduction

Text in news, social media and literature is inherently biased, often in nuanced ways that may perpetuate subjectivity—introducing certain perspectives, presupposing truth and casting doubt. Subjective bias occurs when language that should be neutral and fair is skewed by feeling, opinion, or taste. This kind of bias, while perhaps unintentional, can lead to polarized opinions or unclear portrayals of truth. For example, the sentence “Dave was outed as an underperformer in the office” contains subjective bias that presupposes that Dave is indeed an underperformer. To neutralize this bias, we might replace “outed” with “described” to remove the bias implied by the author, as “described” does not hold subjectivity.

In this work, we explore the use of NLP, specifically a T5 encoder-decoder transformer model, to detect and “neutralize” bias in text by offering a word replacement. A model with the ability to automatically detect biased language and provide fluent, neutral language replacements would be broadly applicable from academia to social media. Such a model could be used as a proofreading tool to check bias in any form of text and minimize either intentional or unintentional subjective biases.

Background

Previous Work

Previous research in bias neutralization has largely focused on decreasing bias in word embeddings that are fed into NLP models, such as in the work of Bolukbasi et al. 2016 [2]. More recent work uses attention-based RNNs [3][4], and BERT-based models to actually edit biased text, which have now become state-of-the-art for bias detection and neutralization [1][5][6].

In particular, we look to the work of Pryzant et al. [1], as they developed the first generative model for our exact problem. In this work, the authors suggest the use of a BERT-based encoder for detection of biased words and an LSTM-based decoder for the generation of neutralized replacement text. A novel join embedding is used to integrate these two separate tasks. In addition to this innovative model, they introduce the Wiki Neutrality Corpus (WNC) [1], a labeled dataset composed of 180,000 biased and neutralized sentence pairs that are generated from Wikipedia edits tagged for “neutral point of view”. This type of edit is guided by principles such as avoiding stating opinions as facts, and preferring non-judgemental language. We use the WNC described here as our own training dataset.

Even more recently, transformer based encoder-decoder models such as T5 [7] have been developed. However, we have not come across an instance of T5 being used specifically for bias detection and neutralization. Here, we propose a T5 model as our architecture to detect and neutralize bias.

Dataset

We utilize a subset of the WNC comprising 53,802 sentence pairs. *Table 1* displays four examples of sentence pairs from the corpus. "Source Text" indicates the original, unedited sentence and "Target Text" the neutralized version. In 52% of sentence pairs from this subset, the "Source Text" contains more words than the "Target Text," indicating that word deletion is a common approach to bias neutralization. The examples in *Table 1* show as such. For example, one edit removes "unfortunately" from the phrase "who unfortunately proved unsuited for the format." Among the remaining sentence pairs, the biased term is replaced with a new word or words. For example, "committed suicide" is replaced with "died by suicide" and "repetitively criticized" with "repeatedly criticized."

Table 1: Sample sentence pairs from the WNC.

Source Sentence	Target Sentence
When he returned, his mother told him his stepfather had *committed* suicide.	When he returned, his mother told him his stepfather had *died by* suicide.
The show's first host was Canadian singer and record producer Jackie Rae, who *unfortunately* proved unsuited for the format.	The show's first host was Canadian singer and record producer Jackie Rae, who proved unsuited for the format.
For this reason the Global Times has been *repetitively* criticized by leftist scholars.	For this reason the Global Times has been *repeatedly* criticized by leftist scholars.
The code for the bot itself is open-source software, and can be *easily* configured to watch for anonymous edits from any IP ranges.	The code for the bot itself is open-source software, and can be configured to watch for anonymous edits from any IP ranges.

Methods

Architecture

For our model architecture, we use T5, a pretrained model developed by Raffel et al. from Google [7]. T5 is a transformer based encoder-decoder model that uses BERT as its encoder and GTP as its decoder. T5 is pre-trained on a multi-task mixture of tasks, for which each task is converted into a text-to-text format. T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task, such as "*Summarize:"* or "*Translate English to German:."* This makes it an extremely powerful text-to-text transformer model.

To address our problem of detecting and neutralizing bias, we fine-tuned T5 on the WNC training set as a form of transfer learning. As such, it is able to take a biased sentence as input and generate a bias-neutralized sentence as output using our custom prefix: "*Neutralize bias:*".

Baseline

We first implemented a baseline model in which we fine-tuned the T5 base model on 500 randomly selected sentence pairs from our

training dataset, for 2 epochs, with a learning rate of 1e-4 and beam size k=3. We found this baseline model to generate sentences in the correct form, and occasionally was able to detect the biased word and neutralize it correctly. The T5 base model was supplied by the HuggingFace PyTorch-Transformers model library, and trained using GPUs available on Google Colaboratory. We used starter code adapted from Roy et al. [11] to fine-tune our T5 model.

Evaluation

To evaluate this baseline and all further models, we ran our model on held out test data (using an 80/20 train/test split) and used four standardized machine translation evaluation metrics, as we are essentially performing a translation from biased to unbiased language. First, we utilize simple Accuracy. If our model's sentence prediction is an exact match to the target, the Accuracy score is 1, if not, the score is 0. Second, we utilize BLEU [8], which evaluates the quality of translated text based on precision of exact word matches. Our third evaluation metric is BLEURT [9], a metric based on BERT that predicts the score a human evaluator would give. Our final metric is BERTscore [10], which scores translation quality using contextual BERT embeddings rather than exact word matches.

Each of these metrics provides a score indicating how similar the candidate is to the reference, with values closer to 1 representing more similar texts. However, each metric measures the translation quality in a slightly different way to help us determine how our model is performing.

To calculate each evaluation metric, we make use of beam search with a beam size k=3. For our final evaluation, we calculate the Accuracy, BLEU, BLEURT, and BERTscore for the best beam prediction for each sentence in the training set. Our model level metrics are calculated by taking the average scores for each metric. In previous research, Accuracy and BLEU are the most commonly used metrics for similar tasks. Current state-of-the-art models on our task by Pryzant et al. achieve model level Accuracy = 0.477 and BLEU = 0.932 [1]. We aim to achieve the highest score possible for each metric.

Results and Discussion

Experimentation

To improve upon our model baseline, we ran a variety of experiments to tweak several of our model parameters and hyperparameters. All experiments and evaluation metric scores can be found in *Table 2*. First, we chose to increase the amount of training data. From our baseline of 500 training sentence pairs, we increased our training set to 10,000 and 50,000 sentence pairs. This simple change resulted in significant increases in Accuracy. Our T5 model trained on 10,000 examples received an Accuracy score of 0.249, up from our baseline Accuracy of 0.060. Even stronger, our model fine-tuned on 50,000 examples received an Accuracy score of 0.360. Additionally, BLEU and BLEURT scores increased. From our baseline BLEU score of 0.717, our model increased to 0.866 and 0.877 with 10,000 and 50,000 training examples, respectively. Similarly, from our baseline BLEURT score of 0.450, our model increased to 0.685 and 0.687 with 10,000 and 50,000 training examples, respectively. By comparison, our baseline BERTscore of 0.965 only increased to 0.986 with 50,000 training examples.

We suspect this is because BERTscore provides a similarity score based on word embeddings. The word embedding for a biased word may be very similar to the embedding for an unbiased word in an otherwise identical sentence because bias is so nuanced. In this case, our baseline model and model fine-tuned on 50,000 sentence pairs may score similarly. In general, we hypothesize that more training examples significantly improves our evaluation

metric scores because simply seeing more examples of this nuanced subjective bias helps the model to learn our task. The growth is roughly logarithmic. This trend is illustrated in *Figure 1*.

Figure 1: Training Size vs. Evaluation Metrics. All models use 2 epochs and a learning rate of 1e-4.

The second experimental approach we took in an attempt to increase our evaluation metric scores was increasing the number of epochs. We found increasing the number of epochs to significantly improve scores on our evaluation metrics. Compared to our baseline model with 2 epochs, running an otherwise identical model on 4 epochs increased the Accuracy score from 0.060 to 0.110 - an increase of nearly 100%. BLEU increased 3% from baseline, BLEURT increased 11%, and BERTscore increased 0.2%. When we ran our best model fine-tuned with 50,000 examples with 3 epochs as compared to 2 epochs, metrics improved across the board. Notably, Accuracy increased from 0.374 to 0.426. Again, we suspect seeing the training data more times is helping our model with this subtle task.

The next experiment we ran was to change the learning rate of our model's Adam optimizer. The baseline model used a 1e-4 rate. A variation of the baseline model with a 3e-4 rate shows significant improvement, increasing the Accuracy score from 0.060 to 0.12 — performing better than any other modification of the baseline model. However, once the model is trained on a larger corpus, the effect of the increased learning rate diminishes and eventually no longer improves Accuracy. With 50,000 training samples and all else held equal, the increased learning rate decreases Accuracy from 0.36 to 0.334. All other evaluation metrics decrease, as well.

An additional experiment we ran was to redefine the task into a multi-stage process that utilized double fine-tuning. We hypothesized that training the model on a classification task to help it recognize bias would result in increased accuracy. We implemented a T5 model using a custom prefix: "*Classify bias*:" that trained the model first on a labeled dataset of 10,000 sentences from our corpus. This selection included the source text from the corpus, which were classified as "biased" and the target text, "unbiased." After training the model on this data, we proceeded with our existing training. On the baseline model, the Accuracy with double fine-tuning improves from 0.06 to 0.08. While this is an improvement, it is not as effective as other experiments at this stage, such

as the number of epochs and learning rate. Once the model is trained on a larger corpus, the effect remains minimal. With 50,000 training examples, Model 8, which utilizes double fine-tuning, performs slightly better in terms of Accuracy than Model 6, which does not. Model 8 achieves an Accuracy score of 0.374 to Model 6's 0.360. However, Model 8's BLEU, BLEURT and BERTscores are all lower.

Table 2: Model Milestones and Scores. The highest score for each evaluation method is in **bold.**

Ref #	Model Milestone	Accuracy	BLEU	BLEURT	BERTscore
0	*Baseline:* 500 training examples, 2 epochs, learning rate of 1e-4	0.060	0.717	0.450	0.965
1	*Baseline*, fine-tuned first on a classification task	0.080	0.706	0.429	0.965
2	*Baseline*, 4 epochs	0.110	0.741	0.502	0.967
3	*Baseline*, learning rate of 3e-4	0.120	0.752	0.538	0.971
4	10,000 training examples, 2 epochs, learning rate of 1e-4	0.249	0.866	0.685	0.985
5	10,000 training examples, 3 epochs, learning rate of 3e-4	0.258	0.862	0.651	0.985
6	50,000 training examples, 2 epochs, learning rate of 1e-4	0.360	0.884	**0.687**	0.987
7	50,000 training examples, 2 epochs, learning rate of 3e-4	0.334	0.804	0.478	0.976
8	50,000 training examples, 2 epochs, learning rate of 1e-4, fine-tuned first on a classification task	0.374	0.877	0.666	0.986
9	50,000 training examples, 3 epochs, learning rate of 1e-4	**0.426**	**0.885**	0.680	**0.988**

Discussion

One major factor influencing our models' performance on our evaluation metrics (and a theme we saw throughout) was the fact that subjective bias is extremely nuanced. For example, the sentence below demonstrates bias the model fails to neutralize.

Generated text: "Bapu, the pilot, and the other passengers miraculously survived unharmed."
Target text: "Bapu, the pilot, and the other passengers survived unharmed."

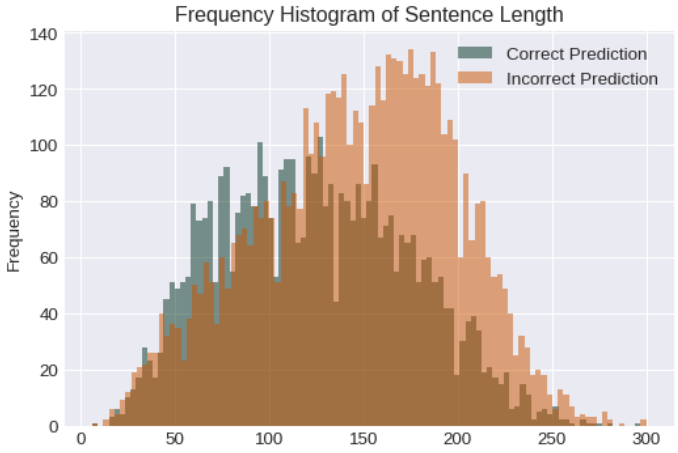
In this case, "miraculously survived" indicates bias and presupposes a level of disbelief towards the event described. However, the model did not recognize "miraculously" as a biased term, and failed to remove it from the sentence. Evaluation of this sentence returns a zero for accuracy, but 0.99 for BERTscore. We believe this is an excellent demonstration of the nuances of bias and the challenges of both building and effectively evaluating a bias detection model. By nature of the edits, the meaning of the sentence should largely remain unchanged before and after neutralization. Instead, the difference in meaning is related to our implicit associations with specific word choice. It is likely that those associations are not captured in word embeddings, making an evaluation metric such as BERTscore difficult to interpret.

Another example can be found in *Table 1*: "For this reason the Global Times has been *repetitively* criticized by leftist scholars" and

"For this reason the Global Times has been *repeatedly* criticized by leftist scholars." To a reader, there may not be a clear bias in the first sentence; "repetitively" and "repeatedly" are synonymous. It is only the connotations of "repetitively" that may imply bias by hinting at superfluousness. Discerning this, and whether it should be neutralized, is complex. We intended to address this problem by first fine-tuning on a bias classification task as described previously, but it was marginally beneficial. Developers of the state-of-the-art bias neutralizing model from Pryzant et al. [1] also acknowledge a similar hurdle in the subtlety of bias, and it seems at current there is no obvious solution to this problem.

Aside from these instances in which our model had a hard time with the nuance of bias, the other cases in which it performed poorly were related to either sentences in which there were several non-English words, or sentences that were long and had many parts separated by commas. The former is due to the fact that T5 was pre-trained on English text, and the latter is likely due to the complex nature and train of thought associated with longer sentences. *Figure 2* illustrates this finding.

Figure 2: Frequency histogram of sentence length (in words), by accuracy.



However, despite these weaknesses of our models, our best model performs very well and the majority of predictions were either an exact match, or a correct neutralization of bias using a near synonym of the target sentence. For example, our model neutralized the input sentence "The film was scored by *jazz great* Herbie Hancock" to "The film was scored by *jazz pianist* Herbie Hancock". This is a valid bias neutralization, however the target sentence is "The film was scored by *jazz musician* Herbie Hancock". While it does not score well on Accuracy, we think this is a perfectly successful accomplishment of our intended task. There are many sentences that follow this pattern.

Overall, computational evaluation of a model like this would probably be most effective if done in tandem with larger-scale human evaluation on a subset of the data. This would likely help researchers better understand the different types of errors the model is making, as well as help discern between true errors in prediction versus errors in evaluation.

Conclusion

Here, we have demonstrated for the first time the viability of using a fine-tuned T5 model to detect bias in text and generate neutral language replacements. After various experimentation, we find our final model to improve significantly over baseline for Accuracy, BLEU, and

BLEURT metrics. Our best model, which takes 50,000 training examples, for 3 epochs, with a learning rate of 1e-4 achieves final Accuracy = 0.426, BLEU = 0.885, BLEURT = 0.680, and BERTscore = 0.988. By comparison, Pryzant et al. [1], developers of the state-of-the-art subjective bias neutralizing model and WNC dataset used here, achieve final Accuracy = 0.477 and BLEU = 0.932. This leaves us within 11% of their best Accuracy score and within 5% of their best BLEU score.

There is still improvement to be made to our T5 model. Several approaches can be taken towards future work. First, more exploration can be made in the vein of specific model architecture. For example, we did not experiment with the number of hidden layers in our T5 model or BERT and GTP subunits or train them separately as a more formal two-stage model. Additionally, we did not incorporate POS tags in our models, although they are part of the WNC and utilized by Pryzant et al.

Overall, our model performs well on its intended task of detecting and neutralizing bias. We hope models such as this one can be used someday as a proofreading tool to make misleading and biased text less prevalent in our society.

References

[1] Pryzant, R., Martinez, R.D., Dass, N., Kurohashi, S., Jurafsky, D., & Yang, D. 2020. "Automatically Neutralizing Subjective Bias in Text". *AAAI*. https://arxiv.org/pdf/1911.09709.pdf

[2] Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings." In Advances in neural information processing systems, 4349–4357.

[3] Christoph Hube and Besnik Fetahu. 2019. "Neural Based Statement Classification for Biased Language". In 12th ACM International Conference on Web Search and Data Mining (WSDM). https://arxiv.org/pdf/1811.05740.pdf

[4] Christoph Hube and Besnik Fetahu. 2018. "Detecting Biased Statements in Wikipedia". In Companion Proceedings of the The Web Conference 2018 (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1779–1786. https://dl.acm.org/doi/pdf/10.1145/3184558.3191640

[5] Tiffany Liu and Tyler Shibata. 2021. "Automatically Neutralizing Ableist Language in Text" Stanford CS224N Custom Final Project https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/reports/final_reports/report079.pdf

[6] Tanvi Dadu, Kartikey Pant, Radhika Mamidi. 2020. "Towards Detection of Subjective Bias using Contextualized Word Embeddings", Association for Computing Machinery, New York, NY, USA. https://arxiv.org/pdf/2002.06644v1.pdf

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. "Exploring the Limits of Transfer Learning with a Unified

Text-to-Text Transformer". Journal of Machine Learning Research 2. 1-67. https://arxiv.org/pdf/1910.10683.pdf

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. "BLEU: a Method for Automatic Evaluation of Machine Translation" ACL. https://aclanthology.org/P02-1040.pdf

[9] Thibault Sellam, Dipanjan Das, Ankur P. Parikh. 2020. "BLEURT: Learning Robust Metrics for Text Generation" ACL 2020. https://arxiv.org/pdf/2004.04696.pdf

[10] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, Yoav Artzi. 2020. "BERTScore: Evaluating Text Generation with BERT" ICLR 2020. https://arxiv.org/pdf/1904.09675.pdf

[11] Shivanand Roy. 2021. "Fine Tuning T5 Transformer Model with PyTorch". https://shivanandroy.com/fine-tune-t5-transformer-with-pytorch/