Gender Bias Detection and Debiasing in Sentiment Analysis

Max Cheever

CMPSC 497: Introduction to Natural Language Processing

Abstract

Bias in natural language processing models can lead to unfair or skewed predictions, particularly in sentiment analysis, where models may associate different sentiment scores with gendered text. This project investigates gender bias in sentiment analysis models and explores methods for detecting and mitigating such bias. Using gender-labeled datasets, convolutional neural networks were trained to classify text by gender and sentiment. The project identified a consistent bias in sentiment classification, where male-associated text received higher positive sentiment scores than female-associated text. A debiasing approach was implemented by adjusting sentiment predictions to balance the positive sentiment ratio between genders while maintaining model accuracy. Results indicate that bias can be reduced effectively without significantly impacting overall model performance.

Contents

1 Problem

1.1 Problem Definition

The goal of this project is to learn how to detect and mitigate gender bias in sentiment analysis models. Sentiment analysis is a common natural language processing task that involves classifying text into positive, negative, or neutral sentiments. However, these models can inadvertently exhibit bias, such as associating certain sentiments with one gender over another. This bias can lead to unfair or skewed predictions, which is particularly problematic in applications like customer feedback analysis, social media monitoring, and automated decision-making systems. All of these things can also further exacerbate existing inequalities within our society, extending to practices such as screening candidates for jobs and curating databases of historical importance. For this project, I chose specifically to focus on gender bias, as gendered pronouns and names provide relatively simple ways to classify sentences as being 'gendered.'

1.2 Dataset Curation

In this project, I chose to use two different data sets: a gender bias evaluation data set and a multiclass sentiment analysis data set from huggingface. The reason I chose these two data sets was because they served different purposes for my project. I used the gender bias evaluation data set to train and test a model that classified sentences by gender, and I used the multiclass sentiment analysis

data set to train and test a model for gender-labeled sentiment analysis. These two models I used in tandem to attempt and detect and mitigate gender bias in my sentiment analysis. This is further explained in the Process Section.

2 Process

You can add small descriptions of what the current sections describe

2.1 Word Embeddings

For my word embeddings, I chose to use the gensim word2vec model. In class, we learned that models such as these are better to use to create word vectors because they create dense vector representations, saving time and space when compared with one-hot vectors. After training this model, I created an embedding matrix where each row corresponds to the embedding vector of a specific word in the vocabulary.

2.2 Algorithm

My algorithm consists of these main parts: implementing a gender prediction and sentiment analysis model, using these models to detect gender bias, and then attempting to debias.

After creating the embedding matrix, I split the data for my gender prediction model into train and test sets, using it to train and then test it. I then trained and tested a sentiment analysis model. For the models, I used CNNs that I defined using the Keras API from TensorFlow. The structure of them is defined as follows:

1. Input layer. This is just a simple sequence of word indices
2. Embedding layer. This layer converts the indices from the previous layer to embeddings.
3. Conv1D layer. Here, I apply 128 filters to the embeddings, each with a kernel size of 5. Each of these filter produces a feature map of size 96.
4. GlobalMaxPooling1D layer. This layer takes the max value from each of the 128 feature maps created by the previous layer.
5. Dense layer. Now, I can create a fully connected layer that outputs a probability between 0 and 1 for binary classification.

After training and testing the gender predictor, I used the gender-predicting model to label each of the sentences for the sentiment analysis data based on gender. I then trained and tested the sentiment analysis model.

After having trained and tested both models, I then attempted to detect gender bias in the sentiment analysis. I did this by calculating the ratio between sentences classified as positive or negative between male and female labels. If I detected bias (i.e. the ratios were different), I calibrated predictions to balance positive sentiment ratios by adjusting a fraction of neutral scores to positive for whichever gender had a lower ratio based on the difference between the ratios. I then re-evaluated both the sentiment analysis model and sentiment ratios to make sure that accuracy had not gone too far down and the ratios were closer (the bias was eliminated).

2.3 Optimizations

In my process, I performed two main optimizations: one for my word embeddings and one for my debiasing.

For my word embedding I originally first trained the word2vec model on just the first

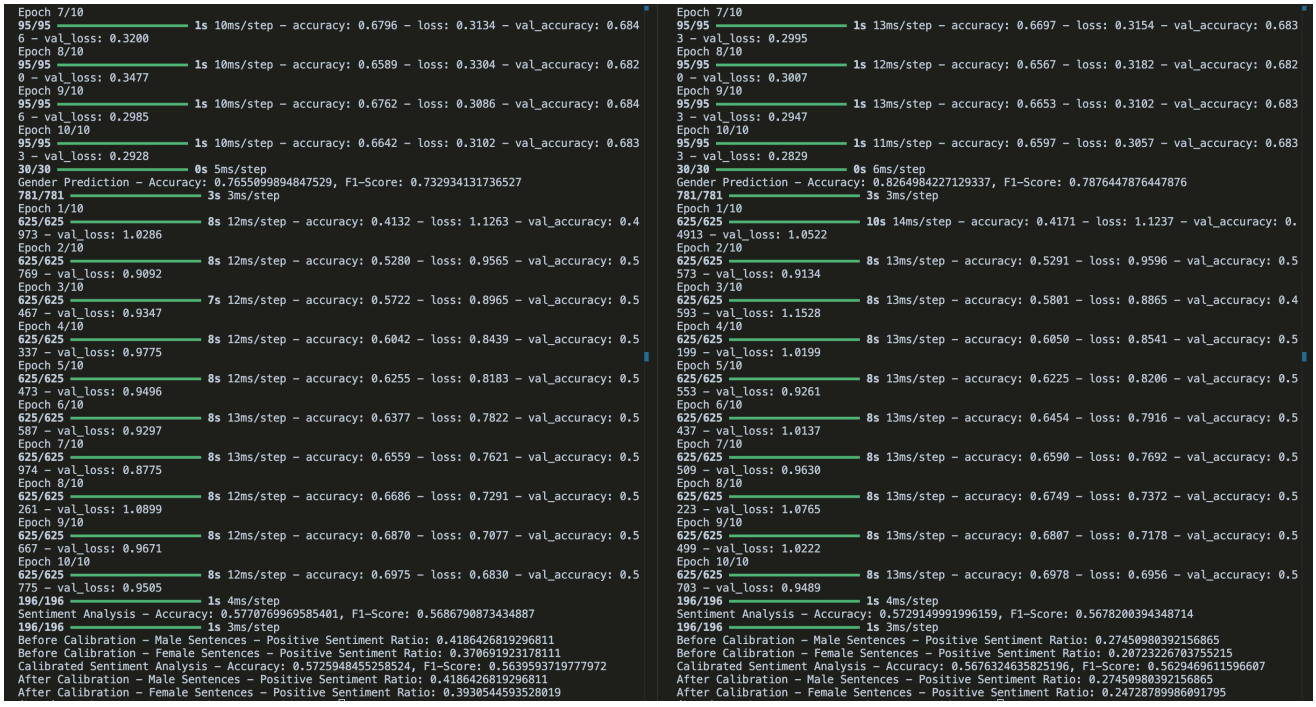
dataset, and then updated it when I trained the sentiment analysis model. This caused issues with the sizes of the training and testing arrays, so I created the entire vocabulary at the same time.

For debiasing, I originally was changing all of the neutral ratings for the gender that the analysis was biased against to be positive, but this created giant swings in the ratios, and also really harmed the accuracy of the model. I figured out that this was just the result of my overcorrecting. I then optimized the debiasing correction to change a certain number of neutral ratings to positive based on the difference between the ratios. This kept the accuracy fairly similar as well as mitigating the bias effectively.

A third optimization that I implemented was to create an easier way for the speech in my sentiment analysis dataset to be classified as 'gendered' as opposed to neutral. What I did was I used a pre-trained tokenizer for parts of speech and replaced all the names in the dataset with gendered pronouns. This did not seem to have an effect on any of my outputs, so I ended up removing it from my process to reduce the computational overhead of my algorithm.

3 Results

Small description



Above I have included a screenshot of two sample outputs from my project. As you can see, in each instance, my gender prediction model does a decent job of assigning gender to gendered speech, and the sentiment analysis model does a little worse. We can also see that the positive sentiment ratio for male sentences is consistently higher than that for female. After my correction, we can see that the ratios pull closer consistently without harming the accuracy of the sentiment analysis model too much.

4 Analysis

Small description

When analyzing the sentiment predictions, a clear bias was observed: male sentences had a higher positive sentiment ratio compared to female sentences. This discrepancy suggests that the sentiment analysis model was influenced by gender-related patterns in the data, leading to unfair predictions. After calibration, the positive sentiment ratios for male and female sentences became more balanced, without harming the accuracy a significant amount. The conclusion that I can draw from this is that there are effective ways to debias sentiment analysis models.

An exploration that I performed during this project was researching debiasing methods. On top of the ones I implemented, I also found more that seemed to be out of scope for this project, including adversarial debiasing. From what I understand, this approach involves training a second model to be an 'adversary' to the original model and then putting them in a debiasing competition.

5 Conclusions

In doing this project, I learned a lot. Firstly, I learned a great amount about a topic in natural language processing that I did not know anything about beforehand. I knew that bias was a large problem in machine learning, but in this project I was forced to learn about more than just the cultural implications. I am now not only confident that I can explain a simple technical approach to debiasing, but I now also fully understand what it means for a machine learning model to be biased. Along with all of this, I also gained a better understanding of convolutional neural networks and their applications within machine learning.