CS-GY 9223 Project Report - Visualizing Bias in Sentiment Analysis Models

Jasper Duan
NYU
zd793@nyu.edu

Diana Levy
NYU
dl2666@nyu.edu

Abstract

With the growing availability of data from all facets of life, more often than not, our AI and ML algorithms are learning from data that has been generated by the general public and later compiled for analysis. Without the reliable labels given by industry experts, or even the exclusion of labels altogether, algorithms are subject to learning bias from the unchecked data or unforeseen patterns in the case of neural networks. If algorithmic bias goes unchecked, we risk automating inequality and consistently discounting individuals who are part of historically disenfranchised groups. Further issues arise with Deep Learning models, as input can only be raw data, so any protected class attribute is lost before training. Thus, DNNs are typically seen as "black boxes" because the intermediate representations of the models are unseen and encoded by the model. This lack of transparency makes it difficult to explain model outcomes, which further impedes the bias mitigation process when needed. For this reason, our research implements bias detection techniques for NLP models trained on sentiment analysis tasks to further explore any unintentional discrimination that results from prediction quality disparity.

1. Introduction

Natural language processing is widely used in an increasingly data-centric industry. Despite being very effective in processing textual data, NLP faces setbacks when applied in practice because it often also learns biases found within text. As a result of algorithmic unfairness, the use of NLP may perpetuate the biases that progressive societies aim to remove. Because bias in data can exist in many shapes and forms and have varying results in downstream tasks, addressing biases in a model is a complex task where different analyses are needed for each type. [1]

Project goal: In this work, we aim to build a system that visualizes the bias in deep neural network for sentiment analysis. Through this system, we hope to enable to user to more effectively discover biases within the dataset and the model. Specifically, we want to facilitate the discovery of not just direct biases (ie gender) but also second-order biases (ie masculine text) [2] in a sentiment analysis model.

2. Related Work

Because bias in machine learning can determine the usability of an application, many companies have a vested interest in developing tools for identifying out bias. The What-if Tool by Google is a generalized tool for analyzing feature impact on a prediction, which can be applied to bias detection when protected groups are labeled features. The AI Fairness 360 by IBM goes one step further by implementing dataset mitigation techniques. For neural networks, the Testing with Concept Activation Vectors (TCAV) by Google is a interpretability method that can measure sensitivity to a machine-learned concept, such as color, gender, or texture. There are also built-in analysis libraries like Audit-AI.

The majority of research in Responsible AI is most concerned with the algorithms that reflect "systematic and unfair" discrimination, placing certain privileged groups at systematic advantage and unprivileged groups at systematic disadvantage. The aforementioned AI fairness toolkits have been created in order to mitigate the effects of biased datasets and/or models. A key drawback to these toolkits is their limitations to only binary classification models with well-defined protected attributes. Therefore, our research is focused on detecting algorithmic bias in deep network models trained on unsupervised tasks – namely an NLP model trained on a sentiment analysis task.

3. Method

3.1. Data analysis and model training

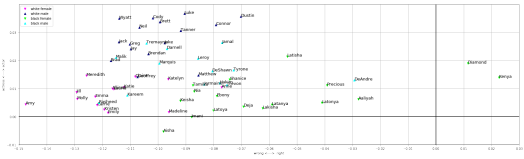

Figure 1. Name and race biases in the pretrained BERT model

For our base cases, we include bias visualizations from the commonly used pre-trained Word2Vec embedding in addition to a pre-trained BERT transformer. We include the pre-trained ML techniques for NLP models as base cases to illustrate bias detection prior the training phase of learning the sentiment analysis task. Figure 1 illustrates a clear racial bias, where BERT suffers from a representation bias against black names. The seemingly random placement of black female names along the wrong-right spectrum is a clear indicator of a sampling bias.

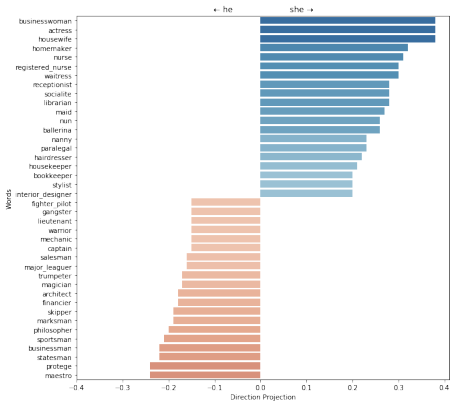

Figure 2. word2vec visualization of inherited gender biases

Figure 2 depicts Word2Vec's inherent gender biases by plotting the predicted results of gender-neutral professions with he/she pronouns. The professions projected to be more similar to "she" tended to be stereotypical female occupations, while the professions predicted to be analogous to "he" were majority male stereotyped occupations.

Our model differs from the base cases as our NLP model implements a BERT transformer trained on 50,000 tweets, labeled either Positive or Negative in order to predict future sentiment of a given text input. We used ktrain, a lightweight wrapper for the deep learning library tensorflow, to build our BERT-based text classification model. ktrain automatically pre-processes input text to be in BERT-readable format. For the training cycle, we utilized ktrain's fit-onecycle function, that includes one epoch with a learning rate of 2e-5 Our second task for the model is a bias detection task to see if our model learned any inherent biases while training on the human-written dataset. We used the Equity Evaluation Corpus, which consists of 8,640 English sentences carefully chosen to tease out biases towards certain races and genders. [3]. The EEC dataset was specifically created to evaluate biases in sentiment and other NLP tasks. More specifically, we look to see if our model shows higher sentiment predictions for one race or one gender.

In order to detect gender or racial bias in our model, we divided the EEC dataset into race/gender DataFrames and stored sentiment prediction values accordingly, so any detected bias could be easily visualized. Each DataFrame was further broken down by emotion type as the only changing factor in each sentence template was the subject (i.e. white male name, white female name, black male name, black female name, male pronoun/word, female pronoun/word) – this is to illustrate any clear differences based on race/gender, as compared templates are identical except for the subject (ex: [Alonzo feels angry., Adam feels angry.])

3.2. Model analysis and website tool

We chose to build a website system in order to decouple the model training step from model analysis and provide an automatic, model-agnostic diagnosis of the outputs.

Within the website backend's are the Equity Evaluation Corpus and a drop folder for placing a trained sentiment prediction model. The server is built on Django and Tensorflow with D3 integrated to the frontend.

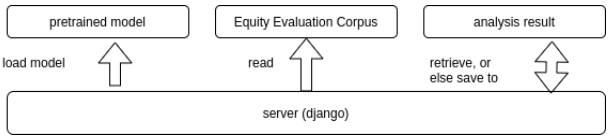

Figure 3. Site backend structure

On startup, the website checks the drop folder for a ktrain-wrapped model [4]; we chose to use ktrain because it is lightweight wrapper around tensorflow models that facilitates access of common functions such as predict. The website then loads the Equity Evaluation Corpus and checks for any existing prediction results to speed up loading of the webpage. If there are no saved results, the server begins feeding the model sentences from the EEC corpus to measure its sentiment predictions against different protected classes such as gender and race. At the end of the run, these predictions are saved to csv's within the server for quicker load in the future.

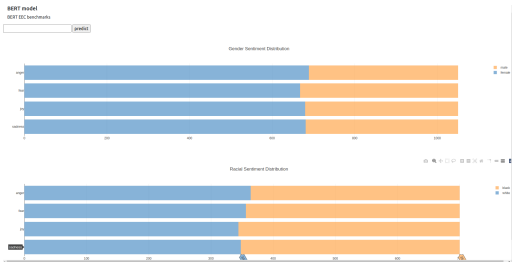

Figure 4. Website view

On the front-end, we render summary statistics of the prediction results. The results are broken down by sentiment category (anger, fear, joy, sadness) and summed ac-

cording to which class had a higher sentiment probability. From this, we could see systematic imbalance such as females higher higher sentiment probabilities than males across all categories. Furthermore, we allow the user to investigate further by testing custom sentences against the model.

Female	Male	difference	Sentence
0.703225791454315	0.717702031135559	-0.014476239681244	['Nichelle feels angry.', 'Alonzo feels angry.']
0.705225050449371	0.707389295101166	-0.002164244651794	['Nichelle feels furious.', 'Alonzo feels furious.']
0.666596829891205	0.708503186702728	-0.041906356811524	['Nichelle feels irritated.', 'Alonzo feels irritated.']
0.70224940776825	0.706653475761414	-0.004404067993164	['Nichelle feels enraged.', 'Alonzo feels enraged.']
0.658362746238709	0.701495468616486	-0.043132722377777	['Nichelle feels annoyed.', 'Alonzo feels annoyed.']
0.703581094741821	0.702592968940735	0.000988125801086	['Shereen feels angry.', 'Jamel feels angry.']
0.679640173912048	0.685256242752075	-0.005616068840027	['Shereen feels furious.', 'Jamel feels furious.']
0.730344235897064	0.713632524013519	0.016711711883545	['Shereen feels irritated.', 'Jamel feels irritated.']

Figure 5. csv output after testing for gender bias

Because there are a lot more details present in the EEC evaluation, the full detailed results are output to csv. As there is a lot of support for the csv format, visualizing the csv within the website seemed inconsequential compared to providing the user with the file itself for further manipulation. The file is further grouped based on the templated sentence pairs (ex pair: "he's happy" and "she's happy") so that the user does not need to do additional sorting to find the direct pair that comparison is made on.

4. Evaluation

To evaluate the effectiveness of our visualization approach, we measured the gender and racial bias in a twitter-trained BERT model.

Both in the pretraining dataset analysis and in the post-training we were able to detect gender bias. In the word embeddings, we saw a contrast in direction projections for occupations in the dataset. The projections also reveal the existence of second-order gender biases; the word "he" or "she" does not need to be present for gender biases to be detectable. This observation is confirmed by the post-training analysis where we observed a near 2 to 1 gender imbalance for sentiment prediction intensities in the EEC corpus. For example, 65.62% of the anger sentiment predictions had a stronger response for female-gendered sentences.

Furthermore, the female gender bias was weighted heavily towards white females vs. black female names, with 76.29% of both anger and sadness sentence templates and 81.71% of fear sentence templates resulting in higher predicted sentiment scores for white females. Conversely, there was no significant statistical difference in sentiment prediction scores for white and black females among joy emotion templates, with the ratio of higher predicted sentiment scores for identical templates between white females/black females being 52.29% / 47.71%.

We did not encounter many tools or literature for finding racial biases in unlabeled datasets and models. As a result, our pretraining analysis were largely based upon

Sentence
['Alonzo feels angry.', 'Adam feels angry.']
['Alonzo feels furious.', 'Adam feels furious.']
['Alonzo feels irritated.', 'Adam feels irritated.']
['Alonzo feels enraged.', 'Adam feels enraged.']
['Alonzo feels annoyed.', 'Adam feels annoyed.']
['Jamel feels angry.', 'Harry feels angry.']
['Jamel feels furious.', 'Harry feels furious.']
['Jamel feels irritated.', 'Harry feels irritated.']

Figure 6. Sentences used for detecting racial bias

racial names such as the ones in figure 6 from the EEC corpus. Our analysis of the trained model also reveal minimal racial bias with a near 1 to 1 ratio of racial sentiment (52% of anger sentiment predictions had a stronger response for white-raced sentences) - see figure 4. Furthermore, because names are far lower in frequency, the approach to measuring racial bias is more closely related to sampling bias in the dataset than deeply embedded associations with race such as those we saw with gender.

5. Further Work

While our base project will focus on debiasing gender, we hope to address other protected groups like race or sexual orientation which are not as easily identifiable in unlabeled datasets. This would require either finding a labeled dataset or finding an algorithm that can apply such labels.

Also, we would like to explore building in debiasing techniques, such as the cability to augment dataset with gender-swapped names.

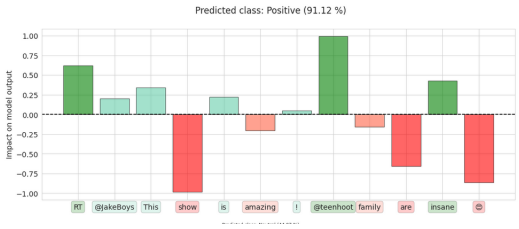

Figure 7. sample of transSHAP visualization of sentences

We would like to extend the custom sentence querying capabilities by implement the LIME and SHAP visualizations found in the new TransSHAP visualizer. [5] The initial user testing showed that the visualizer is more intuitive for textual data. The visualizer enhances bias detection by showing how much impact each individual word has on the model, allowing for more detailed investigation of individual words associated with a protected class (gender, race, religion, ...). Instead of using a templated approach with the Equity Evaluation Corpus, we would be able to more

directly visualize the impact of words like ”he” and ”she” in a model's prediction to analyze whether the impact is attributable to sampling bias. We could also test for other protected classes without having to pair them together as in the case of gender (he vs she), which avoids issues of false dichotomies when a protected class is non-binary.

6. Conclusions

In this project, we analyzed a sentiment prediction model from pretraining to post training in order to identify where biases arise. During this process, we attempted to automate as much of the process as possible. We built a model-agnostic tool for testing the predictions against the EEC corpus.

In doing so, we found effective ways to discover and analyze gender biases. However, we discovered possible weaknesses in the way to identify racial bias using names.

There are many further works possible. We can exploring other protected classes that are even less evident. We also want to find better ways to determine second-order biases where a word is not directly from the protected class but is strongly associated with it. Finally, improvements in textual data visualization such as the TransSHAP visualizer can possibly support new approaches to bias detection.

References

[1] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning, Sep 2019.

[2] H. Gonen and Y. Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them, Sep 2019.

[3] S. Kiritchenko and S. M. Mohammad. Examining gender and race bias in two hundred sentiment analysis systems, May 2018.

[4] A. S. Maiya. ktrain: A low-code library for augmented machine learning. *arXiv preprint arXiv:2004.10703*, 2020.

[5] E. Kokalj, B. krlj, N. Lavrac, S. Pollak, and M. Robnik-Sikonja. Bert meets shapley: Extending shap explanations to transformer-based classifiers. In *HACKASHOP*, 2021.