Measuring and Mitigating Bias in BERT Contextualized Word Embeddings

Suryabrata Dutta

Final Project Submission
DATASCI W266

Abstract

1 Introduction

Word embeddings inherently capture biases from the corpuses they are trained on, and amplify these biases through the methodologies used to construct these embeddings and their implementations in downstream tasks. As word embeddings have evolved in the last 5 years, from GloVe and Word2Vec to contextualized word embeddings like ELMo and BERT, so have the efforts to measure biases (like gender and race) and produce methods to reduce bias effectively. Pre-trained word embeddings are often used as the first layer of processing in many current natural language processing tasks, and contextualized word embeddings are poised to serve as the basis of a wide range of modern NLP applications (most notably, Google's recent decision to switch to BERT as one of the key elements in its search engine (Nayak, 2019)). Therefore, the effort to remove bias from these systems is paramount.

Currently, there is no universal standard of measuring or mitigating bias in word embeddings. The communal work in the last few years have brought forward various approaches to both of these tasks, with varying levels of applicability and success. Most of the effort thus far has been focused on gender bias, more specifically binary female-male gender bias. Recently, more attention is being given to other types of bias (race, religion, etc) as well as applications for multi-class debiasing, but there is still much work to be done in these fields.

Recent efforts have shown promise that embeddings like ELMo and BERT inherently contain less bias by capturing more pertinent information in the context (Zhao et al., 2019). This advantage also comes with a challenge - many of the "word-level" methods of measuring and removing bias are ineffective and/or inapplicable for contextualized embeddings as the actual embedding of each word changes from sentence to sentence. Recent proposed methodologies to measure bias at the sentence level are more effective for this task.

The objective of my work is to implement and modify current methodologies to measure both racial and gender bias in pretrained BERT embeddings, and implement debiasing in the context of a downstream task that builds on / fine-tunes these embeddings.

2 Background

2.1 Measuring Bias in Word Embeddings

Here, I outline two common techniques to measure the level of bias present in word embeddings: analyzing bias subspaces and embedding association tests.

2.1.1 Analyzing bias subspaces

This technique was first introduced in Bolukbasi et al. 2016, where it was applied to measure and remove gender biases in Word2Vec embeddings. First, a list of female-male pairings are generated, such as "she-he", "her-his", "girl-boy" etc. By subtracting the embeddings of the first word from the embedding of the second, and using principle component analysis, the first eigenvector can be generalized to be the "gender direction". The gender bias of non-gendered words (such as specific occupations) can then be measured by evaluating the cosine distance between the word's embedding and the gender direction. Manzini et al. 2019 extends this methodology to a multiclass setting by utilizing n-tuples instead of pairings.

There are a few issues with this approach. First, it requires a large set of handcrafted tuples for each type of bias. These combinations need to be manually crafted, and the final outcomes depend heavily

on the reference sets that the researchers decide to use. This may not matter so much for gender as there are clear pairings that can be fairly generalized (pronouns, family members, etc.). However, this tasks becomes harder for biases like race, where there are less clearly-differentiated analogs (Manzini et al., 2019). The sparse number of comparisons limits the ability to determine the true "direction" of the bias in the embeddings.

Additionally, for contextual embeddings like ELMo and BERT, this presents another issue as the embedding of each word is dependent on context. Basta et al. 2019 solves this problem by using an established corpus, sampling sentences with contain the target word in the pairing, and measuring the difference of the word embeddings by swapping out that word in the given context. This effectively solves the contextual issues, but also adds a dependency of this technique on the corpus they are sampling.

Finally, this calculates the bias for each non-gendered word, but does not give a holistic score or bias p-value for the embedding as a whole.

2.1.2 Embedding Association Tests

These techniques, first introduced in Caliskan et al. 2017, is analogous to the Implicit Association Tests (IAT) which were developed to extract unconscious biases from humans. Generally, these tests contain a list of targets of type A and B (ex: Female and Male names) and lists of attributes X and Y (ex: pleasant words and unpleasant words). It then tests whether or not the association between each target type and each attribute type is uniform across both targets. If it's not, there is a stronger associative bias between the targets and attributes.

Caliskan et al. extends this methodology by introducing the Word Embedding Association Test (WEAT). Given the lists of targets and attributes, it determines the association between each type by calculating the cosine distance between their respective embeddings. One key benefit of this approach is that the analysis states whether the embedding is statistically significantly biases or not (by conducting a permutation test of the attribute pairings). However, a drawback is that like the previous method, it relies heavily on the reference lists of targets and attributes (unlike the subspace method though, it does not have to "pair" certain words).

Contextualized embeddings present the same issue for the WEAT as it did for the bias subspace analysis - the result of the test can differ dramatically depending on the sentence the word is it in. To fix this, May et al. 2019 proposes a new test, the Sentence Encoder Association Test (SEAT), that extends this methodology to the sentence level. To form sentences, May et al. inserts the words used by Caliskan et al. into templates like "This is a[n] $<$ word $>$". These "semantically bleached" as the paper states convey little specific meaning, and doesn't provide much context. While this may be a slight improvement on WEAT, it has the same issue - context is largely ignored.

2.2 Debiasing Word Embeddings

In this section, I discuss two methods of debiasing or mitigating bias: removing bias subspaces and adversarial learning.

2.2.1 Removing Bias Subspace

This technique in Bolukbasi et al. 2016 follows closely with the method detailed above of analyzing the bias subspace. Once the "bias direction" eigenvectors are computed using principle component analysis, word embeddings from gender-neutral words are projected onto the subspace that is defined by the primary eigenvectors. Once this is done, the resulting embeddings are orthogonal to the bias direction (effectively debiasing). This technique has the same challenges as discussed above: it necessitates a handcrafted list of complementary words ("he-she"), and for use in debiasing ELMo or BERT it has to be paired with a corpus.

2.2.2 Adversarial Learning

Zhang et al. 2018 proposed this technique to debias the effect of word embeddings on downstream tasks by actively training against an secondary network during the task learning phase. This method is built on the concept of generative adversarial networks (GANs). During task learning, an "adversary" network is added to the existing "predictor" network, with the objective of predicting the "protected" status (ex. gender or race) of each input from the downstream task predictions. The loss functions are setup in such a way that the adversary network will try it's best to predict, but the predictor network will actively try to decrease it's own loss while increasing the loss for the adversary network. The end goal is a set of predictions in the primary downstream task that is independent of any protected value.

This is an extremely flexible debiasing methodology as it can be applied to a wide range of tasks that utilize word embeddings without the need to train the entire model again (can be used in fine-tuning applications). It is important to note that this may not debias the word embedding directly, and the results do not replicate the same way from task to task. Furthermore, the hyperparameters during the task learning process are extremely sensitive as it is trying to balance loss optimizers for two networks at the same time. As the paper notes, it is usually difficult to ensure stability and convergence.

3 Methods

Code for this project can be found at https://github.com/suryadutta/w266_final_project

3.1 Measuring Bias though Association and Augmentation

From the literature review, it was evident that there was no established method to measure bias in BERT within the context it is used for the downstream task. Therefore in this project, I propose a new variant of the embeddings association test by extending the WEAT/SEAT methodologies. I will refer to this technique as the **Augmented Contextualized Embedding Association Test** (ACEAT).

I chose to use an association test over subspace analysis because there was not a substantial reference for paired words to detect racial bias. The only source I could find was Manzini et al. 2019, which had only 3 pairings.

The main concept of this test is to improve on the SEAT methodology by including context from a corpus with labeled metadata for persons' names. A labeled NER corpus seemed to be a good fit for this effort - I chose to use the **CoNLL 2003 Shared Task** corpus (Sang and De Meulder, 2003). Like the other embedding association tests, I also needed a list of attributes for each bias I was testing. Using Caliskan et al. as a reference, I compiled a list of 40 names divided evenly between two races and two genders (10 African-American female, 10 African-American male, 10 European female, and 10 European male).

The augmentation process is similar to one used by Basta et al. For each time a person's name appeared in the dataset, I replicated the sentence 40 times and replaced the existing name with one from the compiled attribute list. The gender and race of the new name is saved as metadata for use in computation later. As the CoNLL 2003 corpus is pre-split into train/val/test datasets, I did not have to worry about training and testing on the same sentence template.

The association test also requires a target that the attribute is compared to - in Caliskan et al. 2017, they use a list of pleasant and unpleasant words. In lieu of this, I decide to use a semi-supervised methodology and use a pre-trained sentiment classifier to label each sentence as either positive or negative. To avoid any race or gender bias at this step, I anonymize all proper names with a mask "[NAME]" before classifying the sentence. For this project, I opt to use the Flair sentiment analysis classifier (Akbik et al., 2019), an LSTM model trained on the IMDB dataset, but any pre-trained sentiment classifier works for this step.

To compute the bias association for gender or race in the embeddings generated by this dataset, I extend the WEAT algorithm proposed by Caliskan et al.. I compute a test statistic s as follows:

$$s(X, Y, A, B) = \left[\sum_{x \in X} m(x, A, B) - \sum_{y \in Y} m(y, A, B) \right]$$

where m is the difference between the mean cosine distance between the attribute embeddings:

$$m(w, A, B) = [\text{mean}_{a \in A} \cos(w, a) - \text{mean}_{b \in B} \cos(w, b)]$$

In the context of this project, X and Y are the embeddings of the names with opposite attributes (FEMALE and MALE for gender, AFRICAN-AMERICAN and EUROPEAN for race). A and B are embeddings that correlate to the sentiment of the sentence (POSITIVE and NEGATIVE). As the confidence c of each sentiment is given as well, I am able to compute a weighted average in the evaluation of m instead of a simple average. The embedding used for this sentiment is the sum of the embeddings of every word in the sentence *except for the proper name that was placed in during augmentation*. I will refer to this as the "rest-of-sentence" (ROS) vector. The cosine distance inside the calculation of m will only compute the distance of the two embeddings given the same sentence context. Therefore:

$$\text{mean}_{a \in A} \cos(w, a) = \frac{\sum c_i \cos(w_i, ros_i)}{N_A \sum c_i}$$

Once the test statistic s is computed, the statistical significance is calculated via a permutation test:

$$p = \Pr\left[s(X_i, Y_i, A, B) > s(X, Y, A, B)\right]$$

where (X_i, Y_i) is every possible assignment of values in $X \bigcup Y$ to its target.

I also measure the effect size for both race and gender using the following equation from Caliskan et al. 2017:

$$d = \frac{\text{mean}_{x \in X} m(x, A, B) - \text{mean}_{y \in Y} m(y, A, B)}{\text{std\_dev}_{w \in X \bigcup Y} m(w, A, B)}$$

This effect size grows larger as the bias becomes more severe. If both types of biases result in similarly significant results, we can use the effect size to compare the impact of the biases.

3.2 Task Learning

As we can now calculate the bias in the BERT word embeddings, the next step is to see how that bias can shift during fine-tuning in the process of learning tasks, and try to actively debias the results. For this task, I chose Named Entity Recognition as the data corpus I was using already contained these labels.

Of the methods of debiasing discussed above, I choose to use the method of adversarial debiasing as presented in Zhang et al. 2018. I chose not to remove the bias directly through the subspace calculation for the same reason I did not use it to measure bias - there were not enough existing pairing for racial bias.

Using the technique of measuring bias discussed in the previous section, we calculate the bias of the embeddings in three conditions:

1. Before any training (pretrained embedding) - **this is the baseline**
2. After the NER model is trained without debiasing
3. After the NER model is trained with debiasing

My hypothesis is that the **bias will increase from the baseline bias** after the non-debiased training, but decrease from the debiased training.

I will also observe the performance of the adversarial model without and with debiasing from the predictor - I expect to see that the model is performing well without debiasing (indicating that the original NER model is biased in it's predictions), but fail to predict well when debiasing is active. For this analysis, **the baseline will be the frequency of the most-common NER tag**.

3.2.1 NER Model

To prepare the data for NER task learning, I follow the pattern set in the example code provided by the instructors in the W266 class (w266 Instructors, 2019). I start by converting every labeled word in each augmented sentence to their respective tokens. If a word consists of multiple tokens, we attribute the first token to the respective named entity tag and the rest to a "filler" token ["nerX"]. I follow suit with gender and race, adding the attribute to the name tokens. Finally, I add padding at the end of each input to handle variable-length sentences, along with masks. The input ids and masks are then fed into BERT to be evaluated into embeddings, which in turn are directly fed into an output softmax layer for the NER tags to be classified.

I opt not to add layers between the output layer and the BERT layer for two reasons - first, the last four layers of the BERT embedding will also be fine-tuned during training, which will assist in the classification task, and seconds, during bias measurement, we want to capture the "bias" in the embedding itself rather than in layers afterwards which may be more affected during adversarial learning. [1]

I choose to use the **BERT-Base, Cased** model for this project (cased because this is an NER task). All cased BERT models can be used with this methodology.

3.2.2 Adversarial Debiasing

The model described in the previous section was optimized to maximize the classification power of the NER labels by itself. However, bias in this task comes from non-independence between the predictions $\hat{Y}$ and the "protected" attributes (race and/or gender). In other words, these attributes directly impact the performance of the task.

To fix this, we add an additional model on top of the existing one to act as an "adversary". Similar to a discriminator in a traditional Generative Adversarial Network (GAN), the adversary model will take in the predictions $\hat{Y}$ and try to predict these

[1] I chose to fine-tune four layers of BERT as it seemed like a popular option in literature - evaluation bias with various number of layers of tuning is a future direction we may wish to pursue.

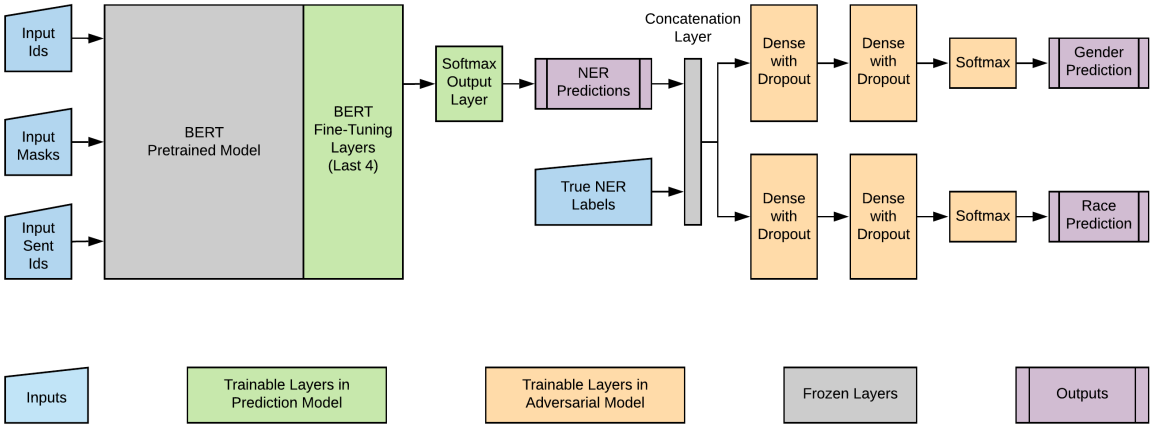

Figure 1: Diagram of the full neural model, including the prediction and adversarial models. The color legend can be found along the bottom.

protected attributes directly. The loss optimizer for this network will act as any normal optimizer and attempt to do it's best as classification. However, for the main network I extend the method that Zhang et al. 2018 establishes, and modify the optimizer for the main network to perform the gradient descent to minimize:

$$\nabla_W L_P$$
$$- \text{proj}_{\nabla_W L_A} \nabla_W L_P - \alpha \nabla_W L_A$$
$$- \text{proj}_{\nabla_W L_B} \nabla_W L_P - \beta \nabla_W L_B$$

The first term, $\nabla_W L_P$, is the typical basic gradient descent optimizer that minimizes the predictor layers' weights ∇_W in a direction that minimizes the predictor loss L_p. The second and third lines are additional constraints that handle the adversarial debiasing for the two protected attributes - gender and race. The loss of these two predictions are labeled L_A and L_B respectively. The first term in the second and third lines serve to prevent the predictor from moving in a direction that would assist the adversary's performance. The second term actively works against the adversary to lower its performance.

It is important to note the two hyperparameters α and β. During the adversarial debiasing process, these parameters are tuned to ensure both the predictor and the adversary can converge. As Zhang et. al. notes, the process of adversarial debiasing is very "touchy" and prone to instability and non-convergence. Tuning these parameters correctly is crucial for the success of this method.

Additionally, I take care to provide the adversary model the true NER labels Y in addition to the predicted labels $\hat{Y}$. This is to ensure a condition of a non-biased algorithm, specifically equality of odds. This condition states that $\hat{Y}$ and the protected attributes must be independent *conditional on* Y.

I also decide to add two large dense layers with dropouts for *each* of the protected attributes the adversary is trying to predict. The rationale here is we want to give the adversary as much of an advantage we can to predict the attributes, so that we can accurately ensure the debiasing works as intended.

4 Results and Discussion

We first look at the results from the ACEAT Test Methodology discussed above, for both gender and racial bias.

In Table 1, we see that the pretrained BERT Embedding result in a statistically significant bias for gender with a 95% confidence interval. However, after training the NER task (with and without debiasing), we see that the p-value increases and the effect size decreases significantly - we fail to reject the null hypothesis that there is no gender bias in the BERT embedding after NER training. Interestingly, the p-values and the effect size don't change significantly before and after the adversarial debiasing step.

In Table 2, however, we see a slightly different story. The pretrained embeddings again show that there is a statistically significant racial bias in the pretrained BERT Embedding. Comparing the gender and racial bias effect sizes for the pretrained embeddings, we see the effect size is much larger for racial bias than it is for gender bias (the p-value

Model	Gender Bias P Value	Gender Bias Effect Size
Pretrained BERT Embeddings	**0.0098***	0.3765
Embeddings after NER Training (No Debiasing)	0.3951	-0.0679
Embeddings after NER Training (With Debiasing)	0.3184	-0.1146

Table 1: Gender Bias ACEAT P Values and Effect Sizes. From the calculations, the targets X and Y refer respectively to FEMALE and MALE associated names, and the attributes A and B refer respectively to POSITIVE and NEGATIVE sentiments. A positive effect size shows a larger bias on average for FEMALE names than MALE names. Asterisks denote a significant result at $\alpha = 0.05$.

Model	Racial Bias P Value	Racial Bias Effect Size
Pretrained BERT Embeddings	**0.00000***	1.1654
Embeddings after NER Training (No Debiasing)	**0.00015***	0.9229
Embeddings after NER Training (With Debiasing)	0.06125	0.3793

Table 2: Racial Bias ACEAT P Values and Effect Sizes. From the calculations, the targets X and Y refer respectively to AFRICAN-AMERICAN and EUROPEAN associated names, and the attributes A and B refer respectively to POSITIVE and NEGATIVE sentiments. A positive effect size shows a larger bias on average for AFRICAN-AMERICAN names than EUROPEAN names. Asterisks denote a significant result at $\alpha = 0.05$.

NER Task Type	Test Accuracy	Test Accuracy - Non Baseline	Gender Accuracy	Race Accuracy
Baseline Metric	0.80	0.58	0.50	0.50
No Debiasing	0.99	0.95	0.50	0.50
With Debiasing	0.99	0.95	0.50	0.50

Table 3: Accuracy scores of models with and without adversarial debiasing. The "Test Accuracy - Non Baseline" column indicates the accuracy of all the tokens except for the one that occurs most frequently. For the normal accuracy, the baseline is $\sim$**80%** (80% of the tokens were labeled as "Object"). For the non-baseline accuracy, the baseline is **58%** (always choosing the second most frequent category - "I-PER"). For both Gender and Race accuracy, the model baseline is **50%** (randomly guessing between 2 categories)

for racial bias is lower as well). However, we also see a racial bias in the embeddings after the initial NER training without the adversarial network involved. The impact of the bias is lessened as the effect size decreases and the p-value increases, but it is still significant. After adversarial debiasing is introduced in task learning, however, the p-value increases to above the specified alpha level, and the effect size decreases as well. It should be noted that this p-value is still much less than it's counterpart for gender bias, and the effect size is much greater. These results show that the **adversarial debiasing method can mitigate some of the racial bias from the embedding during downstream task training**.

These results raise some questions, one of them being "why does the embedding become less biased after the initial non-debiased task training?" To answer this, we look at the model performance from the task training with and without debiasing in Table 3. Here, we see that both of these models perform the NER task extremely well - achieving a 99% accuracy overall and 95% if we omit results for the most frequent label ("Object"). However, even more interesting is the fact that the adversarial network completely fails to predict the gender and race, *even without the adversarial bias steps*. In other words, the predictions that the NER model makes are independent of the race or gender of the name inserted in that sentence. One reason why this may be the case is that the name in the sentence will always have the same true label 'I-PER' regardless of race or gender. Therefore, if the model is as successful as shows here, the results may not be that skewed. Perhaps we would see a higher accuracy in the adversarial network performance (with debiasing disabled) if either the model performed less well, or the labels themselves carried a bias (such as a human-labeled sentiment tag).

Although the adversarial network failed to accurately predict the "protected" attributes at the end of task training, actively debiasing was able to mitigate some of the racial bias that was left during the original training phase. I hypothesize that this is because the active debiasing mitigated bias at the early stages of task training, when model performance may have been skewed, and this impacted the later converged weights.

It is unclear to me why the NER task learning mitigated gender biases from the embedding much better than racial biases.

5 Next Steps

There are a few possible next directions to take from here:

- Test bias and debiasing results with various numbers of fine-tuned BERT layers (this project uses 4)
- Extend the ACEAT methodology to other corpuses, other sentiment classifiers, and to ELMO / other contextualized embeddings
- Test the BERT adversarial debiasing method on more subjective or human-labeled tasks like sentiment analysis
- Pursue other forms of debiasing and compare effectiveness of techniques. This includes bias subspace removal (would require the generation of a large reference of racial words) and corpus debiasing (would need to retrain BERT on a debiased corpus)
- Extend the ACEAT methodology to multiclass / non-binary attributes

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Christine Basta, Marta R Costa-jussà, and Noe Casas. 2019. Evaluating the underlying gender bias in contextualized word embeddings. *arXiv preprint arXiv:1904.08783*.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

w266 Instructors. 2019. Bert ner tutorial. https://github.com/datasci-w266/2019-fall-main/blob/master/materials/Bert/BERT_NER_v1.ipynb.

Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047*.

Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*.

Nayak. 2019. Understanding searches better than ever before.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340. ACM.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. *arXiv preprint arXiv:1904.03310*.