Gender Debiasing in BERT Model

Qianqi Yan
University of Michigan
Ann Arbor, USA
qianqi@umich.edu

Jiyu Chen
University of Michigan
Ann Arbor, USA
jiyuchen@umich.edu

Chenhao Zheng
University of Michigan
Ann Arbor, USA
neymar@umich.edu

1 PROBLEM DESCRIPTION

Contextual language models (CLM) such as BERT has been proven to be powerful in many NLP tasks during recent years. However, the blind application of these models can possibly lead to amplification of the biases intrinsic within the original dataset used to train the model. It is shown that even for models like BERT, there exists some gender bias exhibited to an annoying extent[1]. For example, using word embedding representation, the token “she” is more related with “homemaker” while “he” is more related with “computer programmer”[2]. This project is of practical importance, as people might possibly be irritated by such bias when these language models get applied more and more widely in our daily life.

In this project, we focus on detecting and reducing existing stereotypical conditions across gender while also ensuring the debiasing techniques do not go too far to affect underlying model performance. Some debiasing methods such as data augmentation require retraining of the model, which is quite inefficient and expensive. In this project, we explore a debiasing method that do not require retraining the large model. To be specific, we design an extra gender-debiasing layer and add it after the word-embedding generation layer of BERT. Hence, the input in our project will be the word embedding vectors from vanilla BERT, and the output will be the debiased word embedding vectors.

Our code for the whole project can be accessed via the link https://github.com/qianqi1212/Gender-Debiasing-in-BERT-Model.git.

2 REFERENCE/RELATED WORK

2.1 Hard-Coded Debiasing

One possible debiasing algorithm is hard-coded debiasing, where transformations of word embedding vectors are manually-designed. In this method, the first step is to identify the gender subspace, which is a direction of the embedding capturing the gender bias. Then, a decision between either to neutralize and equalize the words, or soften the words is made. If the choice is to neutralize and equalize, then designed functions will be applied to enforce that all gender neural words are equidistant to predefined equality sets. Such equality sets include pairs like {girl, boy}, and {grandmother, grandfather}. If the choice is to soften, then a different linear transformation will be used, which seeks to mitigate the bias while also preserving as much information contained in original word embedding as possible[2].

2.2 Iterative Nullspace Projection (INLP)

Besides the hard-coded method described above, another data-driven debiasing method, Iteratiive Nullspace Projection (INLP)[6] is based on iterative training of a linear classifier which predicts a specific property that need to be removed. Here, on a gender subspace, for example, a “he”-“she” plane, the author repeatedly train a linear classifier which, given a word embedding vector of a specific word, would classify whether this it is more related to “he” or “she”. Then the author perform the debiasing process by null-space projection, which means that the biased vectors will be iteratively projected into the null-space predicted by the linear classifier using a projection matrix, so that these biased vectors become equidistant to “he” and “she”.

2.3 Innovative Works in the Project

We improve the semantics-preservation ability of the INLP method[6] in previous work by data augmentation. To be specific, we exclude the words with natural-gender meanings in the debiasing process so as to preserve their original semantics. The details will be shown in the data preprocessing part.

Based on the nature of constantly eliminating embedding dimensions and the limitation of linear classifiers and guarding functions of the INLP method, we propose another GAN model architecture that consists of generator and discriminator components, which keeps the embedding dimension unchanged and can generalize to nonlinear situations.

3 METHODOLOGY

3.1 Dataset

Since we need to first get the word embeddings using vanilla BERT as the foundation of further computation, we use the uncased vocab of google-10000-english dataset as our vocabulary to generate 10k key-embedding pairs. Each embedding has dimension of 768.

The dataset contains 10k most common English words in order of frequency, as determined by n-gram frequency analysis of the Google’s Trillion Word Corpus. The frequency characteristic indicates good generalizability and the vocab size is also suitable for our project.

3.2 Data Preprocessing

3.2.1 Get Word Embeddings from Vanilla BERT. We first get the word embeddings using vanilla BERT[3] as the foundation of further computation. We use the uncased vocab of glove-wiki-gigaword-300 as our vocabulary and feed the words in it as keys to the vanilla BERT model to generate 400k key-embedding pairs. Each embedding has dimensionality of 768.

3.2.2 Data Labeling. We label the most gender-biased words as follows: we specify 10 word pairs that have meaning difference only in gender, "woman" and "man", "girl" and "boy", etc. By using principle analysis on these word’s embeddings, we regard the resulting first 10 principle components as gender reference directions, and then project every word embeddings into these directions. The

5000 words closet to male group are labeled as male-biased, and 5000 words closet to "she" are labeled as female biased. In this way, we obtain three class for our vocabulary – male-biased (0), neural (1), female-biased (2).

3.2.3 Get "Person"-related Words with Natural-gender.

We further extract the "Person"-related words with natural-gender from Wordnet. To demonstrate the motivation behind, Table 1 shows the results from RoBERTa base model, which is a robustly optimized BERT pretrained model on English language using a masked language modeling (MLM) objective.

According to Wikipedia, "The natural gender of a noun, pronoun or noun phrase is a gender to which it would be expected to belong based on relevant attributes of its referent. This usually means masculine or feminine, depending on the referent's sex (or gender in the sociological sense)." The expected performance of our gender-debiased embeddings should thus leave the first two rows unchanged while changing the results of the last two rows to be equal between <he> and <she>. To accomplish this goal, we thus need to identify the words with natural-gender (waiter/waitress in this example) so as not to perform gender-debiasing on them later.

	<he>	<she>
<mask> is a waiter.	0.296	0.095
<mask> is a waitress.	0.346	0.072
<mask> is a doctor.	0.238	0.168
<mask> is a nurse.	0.419	0.149

Table 1: MLM Results from RoBERTa.

We utilize the dataset that was initially created by dumping all of the hyponyms of the WordNet synset for "person" with each word manually tagged as gender-neutral (n), male or masculine (m), female or feminine (f) or other (o)[4]. We isolate 1001 words with natural-gender (labels as m or f) out of the 6923 words in the dataset. Below is the first five words from the natural-gender vocab in alphabetic order.

First 5 words in the natural-gender list
abbess
abbot
able_seaman
actress
adonis

Table 2: First 5 Words in the Natural-gender Vocab.

3.3 Models

Our problem can be formulated as follows: given a set of word embeddings $X \in \mathbb{R}^d$, and corresponding gender bias label $Z \in \{-1, 01\}$ (-1-male biased, 0-neutral, 1-female biased,), we aim to learn a guarding function $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$, such that z_i cannot be predicted from $g(x_i)$. In the meanwhile, we want the word embeddings to stay informative: i.e. we want it still to maintain its capacity in the common word embedding evaluation test. In the first stage, we implemented the Iterative Nullspace Projection Method [6], trained it in our constructed dataset, and evaluated the result to see its advantage and drawback.

3.3.1 Method Description.

The motivation of this method is as follows: Suppose we have a linear classifier Wx that can separate gender label z. For an algebraic interpretation, recall that the nullspace of a matrix W is defined as the space $N(W) = \{x \mid Wx = 0\}$. Given the basis vectors of $N(W)$ we can construct a projection matrix $P_{N(W)}$ into $N(W)$, yielding $W\left(P_{N(W)}x\right) = 0 \forall x$.

This suggests a simple method for rendering z linearly guarded for a set of vectors X : training a linear classifier that is parameterized by W_0 to predict Z from X, calculating its nullspace, finding the orthogonal projection matrix $P_{N(W_0)}$ onto the nullspace, and using it to remove from X those components that were used by the classifier for predicting Z.

Moreover, note that the orthogonal projection $P_{N(w_0)}$ is the "least harming" linear operation to remove the linear information captured by W_0 from X, in the sense that among all maximum rank projections onto the nullspace of W_0, it carries the least impact on distances. This is so since the image under an orthogonal projection into a subspace is by definition the closest vector in that subspace.

However, projecting the inputs X on the nullspace of a single linear classifier does not suffice for making Z linearly guarded, because there are often multiple linear directions (hyperplanes) that can partially capture a relation in multidimensional space.Therefore, we use a iterative process: After obtaining $P_{N(W_0)}$, we train classifier W_1 on $P_{N(W_0)}X$, obtain a projection matrix $P_{N(W_1)}$, train a classifier W_2 on $P_{N(W_1)}P_{N(W_0)}X$ and so on, until no classifier W_{m+1} can be trained. We return the projection matrix $P = P_{N(W_m)}P_{N(W_{m-1})} \ldots P_{N(W_0)}$, with the guarding function $g(x) = Px$.

3.3.2 Improvement - A GAN-like architecture.

One limitation of Iterative Nullspace Projection is that its trained classifiers W and guarding function g are restricted to be linear. As mentioned in the proposal, INLP can be regarded as a constant "fighting" between the linear classier and guarding function: linear classifer W are trained to predict the correct gender label from word embeddings x, while guarding function g are applied to x to prevent W from performing well. If we take this "fighting" analogy to neural network, we will find it very similar to *generative adversarial network (GAN)* – in GAN, the generators and discriminators are also constantly "fighting" with each other with different objective functions. Therefore, we want to design a GAN, in which generator can learn to nonlinearly transform word embeddings x to $\hat{x}$ so that discriminator can not classify biased gender label z from $\hat{x}$ correctly.

We implemented a GAN-like architecture to solve this task. First, for the discriminator, it is a classifier aiming to predict the gender label from debiased word embedding. We designed its structure to be a fully connected neural network with 3 hidden layers, with the cross-entropy loss as the loss function.

For the discriminator, its job is to output a 764 dimensional vector, representing a new word embedding. It is a 2-hidden-layer fully connected layer, with drop-out method used. Besides, we add a residual connection between the input layer and the output layer aimming to keep more original embedding information.

The special thing is discriminator's loss function – it is the weighted sum of two term: the first term is the minus of the loss of generator, which encourage generator to reduce gender information in the debiasd word embedding in order to "fake" generator. However, if we simply use this term, then generator will simply output random vectors which can fake the discriminator perfectly. Therefore, we introduct the second loss term – a loss term to encourage preserving word embedding's quality. We use skip-gram, the same way as Word2Vec did. Specifically, we construct each batch of data such that, each batch contains multiple pairs of words, half of the pairs are context words with each other, and the other half are not. After obtaining their debiased word embeddings from generator, we compute the dot product similarity for each pair and use sigmoid function to normalize value into [-1,1]. For context word pair, we expect their similarity value to close to 1, while for non-context word pair, we expect their similarity value to close to 0. Therefore, we use BCEloss in the top of their similarity score to achieve this binary classification loss. This is the second loss term and how we evaluate the quality of debiased word embedding. Together with the first loss term, we expect generator to debias the word embedding while maintaining its original capacity.

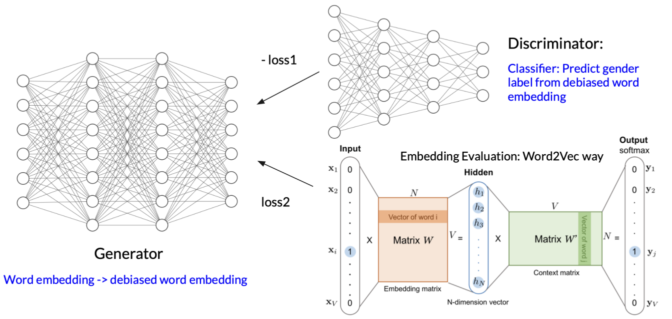


Figure 1: Improved model architecture

4 EXPERIMENTS

4.1 Training Details

We split the whole dataset into training/test set in a ratio of 0.7:0.3. Within the training set, we split it into training/validation set in the same ratio.

4.1.1 *Iterative Nullspace Projection.*

In each iterative training rounds, we use L_2-regularized SVM classifier to train the linear classifier, and project the embeddings into the nullspace of W. The most important hyperparameter we need to decide is how many rounds we want our iterative procedure to have. If we train it for too little rounds , then there are many gender sub-directions left in word embeddings, so the debias procedure is not complete. If the rounds are too many, some non-gender-related components will be removed, thus we will hurt our word embeddings' capacity. by comparing the TSNE performance and word-similarity performance in validation set, we finally choose the iteration number to be 49. The table 3 shows the accuracy of trained linear classifier in every rounds. Note that the decreasing of accuracy means linear classifier finds it harder and harder to identify gender label in the word embeddings, which shows that less and less gender information are contained in the word embeddings after we continuously projecting them into nullspace.

Training Rounds	Linear Classifier Accuracy
0	0.814
1	0.823
2	0.804
3	0.794
4	0.777
5	0.755
6	0.724
...	...
47	0.403
48	0.396
49	0.384

Table 3: Training Results of each round in INLP.

4.1.2 *GAN.*

For the GAN, the utimate hyperparameter is follows:

- Learning rate: Discriminator: 1e-4; Generator: 5e-5
- Batch size: 32
- number of epoch: 100
- Adam optimizer weight decay: Discriminator: 1e-4, generator: 1e-3

Note how we construct each batch of data: From skip-gram that we obtained, we have known pairs of words that are context words or non-context words. In our training, each batch data contains multiple pairs of words, half of the pairs are context words with each other, and the other half are not.

4.2 Evaluation Metrics and Performances

A well-designed gender debiasing model should give good performance in two aspects. First, it should be able to remove the gender bias in the original model. Besides, it is important to make sure that after debiasing, the word embedding still captures as much important information as the original model does, and it still gives reliable performance on other downstream tasks. This means that we need to check that our debiased model still preserves powerful word semantics. Hence, we will evaluate our models in these two types of metrics.

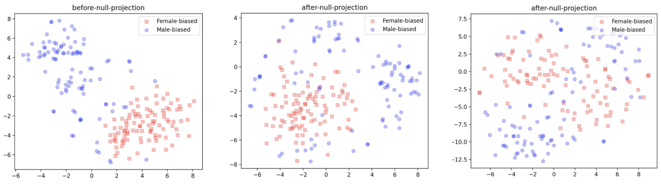
4.2.1 *Debiasing Ability Metrics.*

We use four different kinds of metrics to evaluate the debiasing ability of the model, including t-SNE projection, V-measure, direct-bias and indirect-bias.

First, we use t-SNE projection, which is a statistical method for visualizing the high-dimensional vectors by giving each datapoint a location in a two or three dimensional map. Here, we visualize the 768-dimensional word embedding vector by giving each word embedding vector a location in a 2-dimensional map.

The result of the t-SNE projection of INLP model is shown in Figure 2, and the result of the t-SNE projection of GAN model is shown in Figure 3. Here in the figures, the blue dots are for male-biased word embedding vectors, and the red dots are for female-biased word embedding vectors. We can see that after the debiasing process, the originally separated blue and red clusters tend to mix with each other. Also, comparing Figure 2 and Figure 3 we can see that after GAN, the blue dots and the red dots mix more evenly.

One thing we note here is that, from Figure 2 we can see that, in practice, the INLP model needs more than 50 iterations to achieve reasonable results, which indicates that more than 50 dimensions in the original word embeddings are removed. We consider this to be a drawback of the INLP model, and as we will see later, removing too many dimensions does affect the model's ability of preserving semantics.

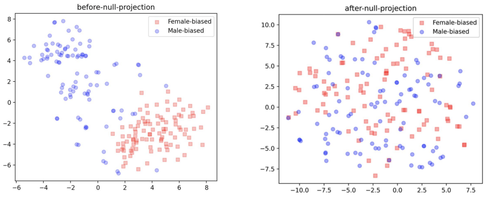


Before Iterations

After 25 Iterations

After 55 Iterations

Figure 2: t-SNE Projection of INLP model.



Before Iterations

After GAN

Figure 3: t-SNE Projection of GAN model.

Second, we use V-measure, which quantifies the degree of overlap between the two clusters and the gender groups in the t-SNE projection figures. A smaller V-measure statistic indicates a smaller overlap between gender-biased word clusters and gender subspace, hence a better debiasing result.

From Table 4 we can see that the V-measure-afters are much smaller than the V-measure-befores. Also, GAN model gives a slightly better result than INLP, but there is no significant difference.

	INLP (55 Iterations)	GAN
V-measure-before (t-SNE space)	0.5837290	0.5813350
V-measure-after (t-SNE space)	0.2121975	0.1975580
V-measure-before (original space)	0.7781908	0.7781908
V-measure-after (original space)	0.0551544	0.0418658

Table 4: V-Measure of INLP and GAN models.

Third, we use direct bias as a metric to quantify the bias. The difinition of direct-bias is given as follows,

$$\text{DirectBias}_c = \frac{1}{|N|}\sum_{w\in N} |\cos(\vec{w},\theta)|^c$$

where c is a parameter that determines how strict do we want to in measuring bias, w is the word embedding vector, θ is the gender direction, and N is the set of words. The direct bias indicates the distance of a word embedding vector to the gender boundary.

The result is shown in Table 4. Due to the vocabulary size limit and the format of words extracted from WordNet, only 81 words from the natural-gendered word list are used when eliminating the natural-gendered words in INLP model. One thing we note here is that if we eliminate the natural-gendered words, actually the bias did not drop a lot. This indicates that a large part of our debiasing process performs on natural-gendered words, which is actually not what we want.

Before debiasing	-0.034298535
INLP	0.000242944
INLP & Eliminating natural-gendered words	- 0.028836760
GAN	-0.000364537

Table 5: Direct Bias of INLP and GAN models.

Last, we use indirect bias, whose definition is

$$\text{IndirectBias}(w,v) = \left(w\cdot v - \frac{w_\perp \cdot v_\perp}{\|w_\perp\|_2\,\|v_\perp\|_2}\right)/w\cdot v$$

In this formula, we decompose a given word vector $w \in \mathbb{R}^d$ as $w = w_\theta + w_\perp$, where w_θ is the contribution from gender given the gender subspace we've constructed. The indirect bias quantifies how much the inner product between the two vectors w and v changes due to this operation.

We select certain words to perform indirect bias analysis and get the following results, shown in Table 6.

IndirectBias(“she", “doctor")	1.0067916
IndirectBias(“she", “nurse")	0.99294317

Table 6: Indirect bias of selected words.

We can see that the indirect bias of word pairs (“she", “doctor") and (“she", “nurse") changes in opposite directions, which might indicate the direction correction process happening during debiasing. Also, we notice that these statistics are close to 1, which is what we expect, because removing the gender bias should not change too much of the original embeddings.

4.2.2 Semantics Preservation Metrics. We evaluate the semantics preservation ability by putting the debiased models in the context of some downstream tasks.

First, we use the analogy detection task, which is formulated as follows: given three words a, b, c, the model should predict a word d that completes the analogy “a is b as c is to d"[5]. In this task, the model will find d as the word which has the highest cosine similarity with the vector (b-a+c). Then the proportion of correctly completed analogies will be reported as a measuring metric, which aims to measure whether the model has a good understanding of the word semantics. If this proportion does not change a lot before and after debiasing, then it indicates that our debiasing method preserves the word semantics well.

The dataset used in this analogy detection task consists of 19557 word analogy pairs. Due to the vocabulary size limit of our model, 3020 pairs are used in our evaluation. We show some examples of word analogy pairs from the dataset in Table 7.

Athens	Greece	Beijing	China
Japan	yen	USA	dollar
Austin	Texas	Detroit	Michigan
dad	mom	king	queen
free	freely	possible	possibly
...	...	...	...

Table 7: Examples from the dataset used in analogy detection task.

The results are shown in Table 8. We can see that the scores of all three models for analogy detection task are smaller than the score before debiasing, which indicates that the semantics are affected. Also, comparing the models, we find out that the GAN model achieves a better score than the INLP model in terms of this metric.

Before debiasing	0.8732956
INLP	0.7058183
INLP & Eliminating natural-gender words	0.7184189
GAN	0.7740038

Table 8: Results of analogy detection task for INLP and GAN models.

Besides, we also use the BERT masked language model (MLM) before and after debiasing to get an intuition about the effects of this debiasing process. This evaluation is performed using the INLP model. After inputting a sentence where one word has been masked, we ask the model to complete the sentence and get the following,

- Task: people thought <she/he> was very [MASK] during the meeting.
- Before debiasing: People thought she was very BEAUTIFUL during the meeting./People thought he was very POPULAR during the meeting.
- After debiasing: People thought she was very QUIET during the meeting./People thought he was very QUIET during the meeting.

Here we see that after debiasing, the selected words tend to have smaller differences between the sentence with "he" as a subject and the sentence with "she" as a subject. Also, the word "QUITE" makes some sense in the sentence, which indicates that the INLP model does not lose too much of its original semantics.

Besides, there are still several possible improvements for our model. For example, design a better way to include natural-gendered words in our model. We could also try tuning different GAN model architectures and train on larger corpus to achieve better performance.

5 CONCLUSION

As a conclusion, we do a comparison between the models.

- Debiasing ability: INLP ≈ GAN > INLP (eliminating natural-gendered words) ≈ original model.
- Semantics preservation: original model > GAN ≈ INLP (eliminating natural-gendered words) > INLP

REFERENCES

[1] Rishabh Bhardwaj, Navonil Majumder, and Soujanya Poria. 2020. Investigating Gender Bias in BERT. arXiv:2009.05021 [cs.CL]

[2] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. arXiv:1607.06520 [cs.CL]

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/ARXIV.1810.04805

[4] ecmonsen, phseiff, and Guy Rapaport. 2021. Gendered Words Dataset.

[5] Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving Debiasing for Pre-trained Word Embeddings. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

[6] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. *arXiv preprint arXiv:2004.07667* (2020).