Reducing Sentiment Polarity for Demographic Attributes in Word Embeddings using Adversarial Learning

Chris Sweeney
MIT
Cambridge, MA
csweeney@mit.edu

Maryam Najafian
MIT
Cambridge, MA
najafian@mit.edu

ABSTRACT

The use of word embedding models in sentiment analysis has gained a lot of traction in the Natural Language Processing (NLP) community. However, many inherently neutral word vectors describing demographic identity have unintended implicit correlations with negative or positive sentiment, resulting in unfair downstream machine learning algorithms. We leverage adversarial learning to decorrelate demographic identity term word vectors with positive or negative sentiment, and re-embed them into the word embeddings. We show that our method effectively minimizes unfair positive/negative sentiment polarity while retaining the semantic accuracy of the word embeddings. Furthermore, we show that our method effectively reduces unfairness in downstream sentiment regression and can be extended to reduce unfairness in toxicity classification tasks.

CCS CONCEPTS

• **Computing methodologies** → *Natural language processing*.

KEYWORDS

embeddings, fairness, NLP

ACM Reference Format:
Chris Sweeney and Maryam Najafian. 2020. Reducing Sentiment Polarity for Demographic Attributes in Word Embeddings using Adversarial Learning. In *Conference on Fairness, Accountability, and Transparency (FAT* '20), January 27–30, 2020, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3351095.3372837

1 INTRODUCTION

Sentiment analysis is used in plenty of applications to mine text data. Rating movies, ranking restaurants by reviews, censoring toxic online comments, etc. all benefit from tools that can infer sentiment from written text. The creation of word embeddings through algorithms like word2vec [16] has led to increased performance in NLP tasks like sentiment analysis, but has also served as a new vector for unintended bias. Recently, works like [5, 6] have shown that word embeddings can contain many types of unfair biases related to gender and race. In this work, we focus on word embeddings that contain unintended bias due to demographic attributes having unfair polarization towards positive or negative sentiment. A demographic attribute is a loose concept where the exact definition can vary between applications. We focus on demographic attributes in the form of demographic identity terms. We define a demographic identity term as a one word descriptor that can be used to assign a person to a particular demographic. This can range anywhere from national origin terms like *American, Mexican*, religious identifiers like *Catholic, Jewish*, gendered words such as *Female, Male*, or even names that tend to belong to African American demographics like *Darnell, Lakisha*. The unequal treatment of demographics via such textual demographic attributes is concerning given how entangled sentiment analysis is with domains that directly impact society. As studied in [19], biased word embeddings can have adverse consequences when deployed in applications such as movie sentiment analyzers or messaging apps.

In a perfectly fair scenario, we argue that demographic identity word vectors should be neutral with respect to sentiment. We refer to the unfair distribution of sentiment among demographic identity terms as sentiment bias or sentiment polarity. Unfortunately, unlike gender bias, which has been thoroughly explored in the research community, sentiment bias is a fairly loose concept. Consequently, it is a much more difficult problem to decorrelate word vectors with sentiment, while retaining the usefulness of the embeddings. However, adversarial learning techniques have recently proven to be a powerful tool for reducing unintended bias in machine learning [2, 8, 23, 28]. We reduce sentiment bias at the word embedding level, using adversarial learning to decorrelate demographic identity term word vectors with sentiment, thereby, removing sentiment bias at its source.

Unintended bias, especially in NLP, can enter a machine learning application through many different sources (i.e word embeddings, dataset, choice of machine learning algorithm, etc.). However, due to the ubiquity of pre-trained word embedding models used in NLP, mitigating bias at this level has the potential for a very large impact. Focusing in on sentiment analysis applications, practitioners need to have control over the possibly unfair sentiment associated with demographic identity terms, without losing the semantic meaning of the vector space. A tool that decorrelates word vectors with sentiment is very general and would enable the practitioner to re-embed chosen words without putting impedance on the practitioner's downstream algorithms. This is beneficial over further downstream debiasing algorithms that require the practitioner to change the learning algorithm or classification thresholds.

This paper starts by reviewing related works in Section 2. In Section 3, we formalize the notion of sentiment polarity and present our adversarial learning algorithm for debiasing word vectors with

FAT '20, January 27–30, 2020, Barcelona, Spain*

ACM ISBN 978-1-4503-6936-7/20/02…$15.00
https://doi.org/10.1145/3351095.3372837

respect to sentiment. Section 4 evaluates the effectiveness of our method. We show that our method increases fairness for sentiment applications and even toxicity prediction applications via benchmarks created in [14] and [7]. In Section 5, we discuss possible future directions for our work. Finally, we conclude this work in Section 6.

2 RELATED WORK

Unintended bias in NLP is often subtle and can emanate from many different sources. Many researchers have focused on understanding the forms and sources of unintended bias in standard NLP systems [3, 12, 25]. Researchers have also made progress towards formally measuring and mitigating some of this bias at different stages in the NLP pipeline. Starting with word embeddings, works like [5, 6, 10, 24] have made progress towards identifying and mitigating many different types of bias with respect to gender and race. However, identifying and mitigating unintended bias is still a vast and complicated problem as [11] argues. Further downstream from word embeddings, researchers have made progress in preprocessing text datasets to mitigate unintended bias in classification algorithms. [7, 21] create methods to balance the training set with respect the portion of identity terms appearing in toxic or non toxic contexts. [9] uses adversarial learning to obfuscate demographic information about the author given written text. This idea is similar to our method, however, we seek to use adversarial learning to decorrelate sentiment bias for the content of the text itself.

Finally, there have been some efforts to create benchmarks for fairness in various NLP tasks. For sentiment analysis, [14] creates a corpus of 8,640 English sentences called *Equity Evaluation Corpus (EEC)* used to manifest bias with respect to race and gender. They examine over 200 sentiment analysis algorithms using this benchmark and show that many algorithms have different sentiment intensity predictions for sentences with different demographic identity terms. We use the same benchmark to show that algorithms trained with our debiased word embeddings are fairer with respect to sentiment bias. Additionally, [7] uses a similar benchmark to uncover unintended bias in toxicity classification systems. We use this benchmark to show that our debiasing algorithm is general and can be applied to other domains like mitigating harmful bias in toxicity classification.

3 METHODS

We present an adversarial learning method to decorrelate word vectors with sentiment. Our algorithm is trained on a large corpus of words, but only chosen protected attributes are re-embedded with the model's debiased predictions. In presenting our approach, we first describe sentiment polarity in terms of vector subspaces. Then we present our adversarial learning algorithm that debiases word vectors with respect to these subspaces.

3.1 Creating the Sentiment Polarity Subspace

Since word embeddings are not explicitly labeled with the dimensions that correspond to sentiment, we must define subspaces that capture negative or positive sentiment and assess word vector sentiment polarity by some distance metric to each subspace. [5] and

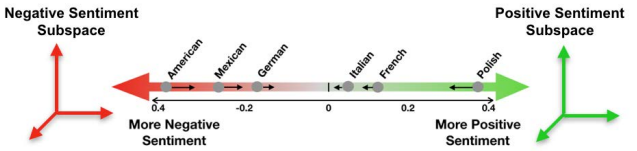

Figure 1: Demographic identity terms naturally have different sentiment polarity within an embedding space. Our goal is re-embed identity term word vectors with as little sentiment polarity as possible, without distorting their semantic meaning. By decorrelating word vectors with sentiment direction, we can move all identity terms towards a neutral point with no projection onto sentiment direction.

[28] use a similar method to investigate gender bias that pairs gendered words like male and female, and extracts a directional vector from the PCA decomposition from a set of these pairs. We tried this approach, but unfortunately it does not work well for sentiment as the semantics for what defines a positive or negative concept is much looser than for gender. Additionally corresponding positive and negative words do not have as clear pairwise mappings as gendered words (ex. male:female). To mitigate these two issues, we first take the most significant PCA component of a matrix of positive word vectors, and do the same with negative word vectors from the Sentiment Lexicon dataset [13]. We then take the signed difference between these positive and negative principal components. We found that the first significant component contains significantly more variance than any other component. Figure 2 displays the top 30 principal components of the positive sentiment word matrix (left), and negative sentiment word matrix (right). One can see that most of the signal for capturing positive or negative sentiment can be captured in the first principal component. We call the signed difference between these positive and negative principal components our *directional sentiment vector*, as it connects the positive and negative subspaces. Given a word vector, we can now project it onto the directional sentiment vector to assess its sentiment polarity. A visualization of the resulting sentiment polarity for some national origin identity terms is displayed in Figure 1.

For our experiments, the more positive the projection, the more positive sentiment polarity. More negative projections correspond to more negative sentiment polarity. The exact reasons why identity terms have differing sentiment polarity is not clear, but the effect of this inequality in downstream algorithms can cause discrimination. We choose to equalize the sentiment between identity terms by decorrelating the word vectors with sentiment, pushing the identity terms closer to 0, or in vector terminology, minimizing the projection onto the directional sentiment vector. One can naively accomplish this goal by projecting all word vectors onto our sentiment subspace. However, this projection distorts the word vectors beyond usability. We instead use an adversarial framework to balance word vector distortion with sentiment bias. We explore this tradeoff in our experiments.

3.1.1 Validating the Directional Sentiment Vector. We need to validate that the directional sentiment vector effectively captures sentiment polarity to ensure we can effectively decorrelate word vectors

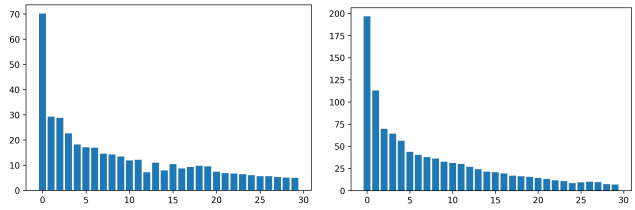

Figure 2: Top 30 principal components of the positive sentiment word matrix (*left*), and negative sentiment word matrix (*right*). In both cases, the first principal component contains much of the signal for representing positive and negative sentiment in the embedding vector space. This suggests we can use the first principal components to represent much of the semantic content for the positive and negative sentiment subspaces.

with sentiment. Therefore, we measure the accuracy of the following classification model.

$$\hat{y}^* = sign(k^t x_i^*)$$

where x^* is a word vector and y^* is a positive or negative label from the Sentiment Lexicon dataset [13]. The Sentiment Lexicon is an established corpus for positive and negative sentiment word labels used in many works [15, 20, 26]. The resulting precision of the classifier, $\hat{y}^*$, on this lexicon was 91.3%. This assures that the directional sentiment vector, k, contains much of the necessary information to determine sentiment polarity. To further prove this notion, we later show that word vectors that have less sentiment polarity are fairer when used in downstream sentiment valence regression tasks.

3.2 Adversarial Learning to Reduce Word Embedding Sentiment Bias

We use an adversarial training regime to subtract out the unwanted sentiment correlations from our word vector. We define our depolarized sentiment vector, $\hat{y}$, as $\hat{y} = y - ww^T y$, for some learned weights w, and possibly sentiment biased word vector, y. To train and find weights, w, we define two competing objectives. Firstly, we want to make sure that the adversarial objective does not distort the meaning of the word vector. Since the word embedding vector space is not very interperatable, there does not exist an obvious loss function that encodes word vector distortion. Therefore we describe a simpler loss function, L_p, that minimizes the mean squared distance between our input word vector and debiased word vector, $(y - \hat{y})^2$. Conversely, for our adversarial objective, we want to pull y away from the positive and negative sentiment subspaces. We define an adversarial objective, L_a describing the ability for an adversary to predict the polarity of the word vector. As described in the previous section, sentiment polarity is defined by the projection of a word vector onto the directional sentiment vector, k. The adversary therefore tries to predict the sentiment polarity, z from the input word vector, $z = k^t y$. The adversary prediction problem is, $\hat{z} = w_a^T \hat{y}$, for adversarial weights w_a, learned with mean squared distance loss, $(z - \hat{z})^2$ or L_a. We combine L_a and L_p in a gradient update for weights, w, using a similar methodology to [28]. We minimize the following objective.

$$\nabla W L_{p_w} + proj_{\nabla W L_a} \nabla W L_p - \alpha \nabla W L_{a_w}$$

The middle projection term ensures that L_p does not end up helping our adversary as described in [28]. Furthermore, α is a parameter used to trade off the distortion of the semantics of the word vector and the fairness via sentiment bias. We investigate this trade off in our experiments.

4 EXPERIMENTS

We evaluate the effectiveness of our method in minimizing the sentiment polarity in a word vector without distorting the word vector's semantic meaning within the vector space. We first evaluate how we can decrease sentiment bias in the depolarized word embeddings for sets of identity terms. Next, we show how our word vectors are not distorted when decreasing sentiment bias. Finally, we take a look at how our debiased word embeddings make a downstream machine learning task fairer. We evaluate two case studies: one in sentiment valence regression and one in toxicity prediction.

4.1 Datasets for Experiments

For our experiments, we need a set of positive and negative sentiment words to create our sentiment subspaces. We use the Sentiment Lexicon dataset from [13] to supply the necessary words. The word embedding models that we debias are the word2vec [17] pre-trained model trained on a large corpus of Google News data and GloVe word embedding [22] trained on a large Wikipedia corpus. Recent studies have shown that these pre-trained models contain many types of bias [4, 6]. The focus of this work is on minimizing the sentiment bias contained in the embeddings with respect to demographic identity terms. We evaluate our method on various sets of identity terms presented in previous works [7, 14]. These sets are by no means exhaustive, however, we show that we can reduce sentiment bias for many different types of identity terms. To evaluate word vector distortion before and after debiasing, we use the WordSim353 similarity dataset developed by [1]. It is difficult to comprehensively evaluate a word embedding model, but correlations between word vector similarities and human-assigned similarity judgments from WordSim353 gives us an insight into accuracy of our debiased word embeddings. For our downstream application, we investigate how our debiased embeddings make fairer sentiment valence regression models trained on the dataset from *SemEval-2018 Task 1 Affect in Tweets* [18]. We also investigate how our debiased embeddings make fairer toxicity classification models trained on the Wikipedia Talk dataset [27].

4.2 Reducing Sentiment Bias in Word Embeddings

We trained our algorithm on the first two million words from the word2vec embeddings and four hundred thousand words from GloVe to find the best way to reduce sentiment bias for each model. The use case of our algorithm is to re-embed words with our trained model that should not have sentiment polarity like demographic identity terms. We evaluate the sentiment polarity of demographic identity terms before and after debiasing.

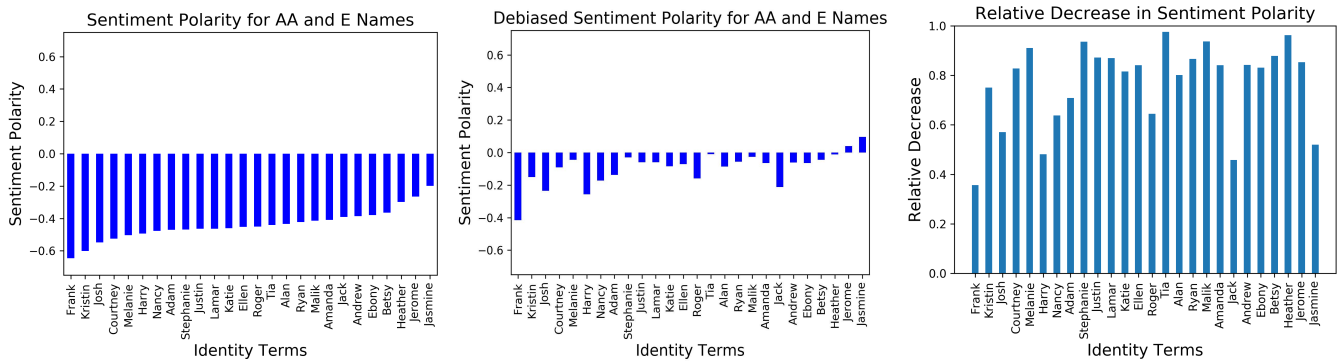

Figure 3: Histograms showing sentiment polarity for names that tend to belong to African American (AA) and European (E) demographics. The left histogram shows the sentiment polarity for the demographic identity terms before debiasing. The middle histogram shows the demographic identity terms after debiasing with our method. The histogram to the far right shows the relative decrease in sentiment polarity after debiasing the identity term word vectors. After debiasing, we can reduce the overall sentiment polarity for this type of demographic attribute.

Demographic attributes can be represented in text in many ways (i.e gendered words like *he, she*, religious identifiers like *Catholic, Jewish*, or even names that tend to belong to certain race. We can effectively reduce sentiment bias for many demographic attributes in text. A quick way to measure sentiment bias is to sum the absolute value of the projections onto the sentiment vector for demographic identity terms within a group of demographic attributes. We call this sum, the *summed sentiment polarity*. Table 1 shows the relative decrease of *summed sentiment polarity* for three different types of demographic attributes: gendered terms, names, and religious identifiers. The names and gendered terms come from those used in [14] to study sentiment bias and the religion identity terms we use are chosen from the most popular religious identifiers in the world. It is important to note that in a real world use case, a practitioner would pick the exact terms to debias using our algorithm. The lower the *summed sentiment polarity*, the more effectively we can remove sentiment bias for that type of demographic attribute. We examine how this lower sentiment bias actually makes downstream sentiment analysis algorithms fairer for names and gendered words in later experiments.

	Gender	Names	Religions
Relative Decrease in Sentiment Bias: GloVe	79%	53%	45%
Relative Decrease in Sentiment Bias: Word2Vec	59%	76%	63%

Table 1: Table showing relative decrease in summed sentiment polarity for identity terms for three types of demographic attributes: *Gender*, *Names*, and *Religions* after applying our algorithm. For popular pre-trained models GloVe and Word2Vec, we can effectively reduce sentiment bias within the embedding vector space.

Figure 3 displays histograms showing the projection of identity term word vectors describing typical names from European and African American demographics onto the directional sentiment vector, before and after debiasing. The resulting number is the sentiment polarity for a given word vector. We also show the relative decrease in sentiment polarity in the far right plot. Although the decrease sentiment bias is not equal for every term, we still attain an average relative decrease across the names of 76.8%. This makes it much less likely for a downstream sentiment analysis algorithm to pick up on the sentiment bias. It is also worthwhile to note that most demographic identity terms tend to naturally be polarized towards the negative subspace. This is however irrelevant given that our goal is to re-embed word vectors without positive or negative sentiment polarity.

Furthermore, we want to verify our models' ability to generally reduce sentiment bias for new words, not seen in the training set. To this end, we removed the identity terms used in this experiment from the training set. We trained our models for 40,000 steps with batch size of 1000 words. The models were trained with an adversarial weight, $\alpha = .5$.

4.3 Post-Debiasing Word Embedding Performance

$K = 10$ **Nearest Neighbors Before Debiasing**	$K = 10$ **Nearest Neighbors After Debiasing**
male: 1.000	**male**: 1.000
female: 0.840	**female**: 0.817
males: 0.758	**males**: 0.733
females: 0.703	**females**: 0.669
accomplice-Hudgens: 0.636	**Male**: 0.602
Male: 0.629	**Female**: 0.565
Female: 0.597	**femal**: 0.543
femal: 0.585	**Caucasian-male**: 0.522
Caucasian-males: 0.557	**accomplice-Hudgens**: 0.517
masculinised: 0.534	**heterosexual-males**: 0.491

Table 2: Top 10 nearest neighbors for the demographic identity term, *male* (bolded), before and after debiasing this word using our technique. There is little distortion of the word vector's relationship to its neighbors after decorrelating with sentiment.

As our model moves word vectors around in the embedding vector space, we also run the risk of distorting the vectors, possibly

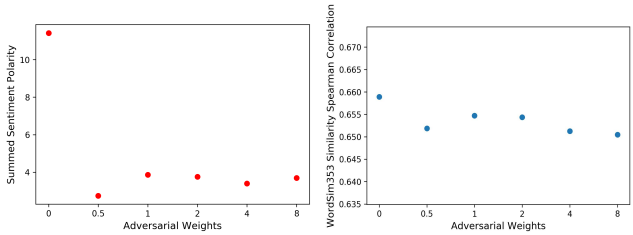

Figure 4: *Left*: Plot summarizing how sentiment polarity for the set of names presented in Figure 3 varies for different adversarial weights. *Right*: Plot showing how the WordSim353 word embedding performance varies with different settings of our adversarial weight.

losing their semantic relation to the words around them. To evaluate the debiased word embeddings with respect to their relation to other words, we can use notions like analogy completion tasks or similarity measures to surrounding word vectors. After debiasing word vectors like *man*, we still retain the word vector analogy, man:woman as boy:girl. Furthermore, the nearest neighbors for a particular word vector are almost identical before and after debiasing. Table 2 shows the top 10 nearest neighbors (measured with cosine distance) for the demographic identity term, *male*, before and after debiasing using our technique. The order of the nearest neighbors and their relative distances, are barley changed. This suggests that our algorithm does not need to move word vectors far to effectively decorrelate with positive or negative sentiment, rather, it finds the right directions to move them to decorrelate with sentiment.

We also analyze our debiased embeddings more formally on the word similarity dataset, WordSim353 [1]. The dataset is composed of pairs of words with labeled human similarity judgments. Using our trained model, we debias one word from each pair, evaluate the resulting cosine similarity and compute spearman correlation with the human judgments in WordSim353. The spearman correlation serves to measure the effect debiasing one word has on its semantic relation to another word from the embeddings. We evaluate spearman correlation score for 6 different models trained with differing adversarial weights, α, and compare to our the *Summed Sentiment Polarity* for the set of names presented in Figure 3. Figure 4 shows the spearman correlation vs sentiment bias for 6 settings of α. The spearman correlation is barely changed by debiasing the word vectors. This is likely due to the fact that our loss function L_p constrains a debiased word vector from straying too far away from its original place in the vector space.

4.4 Downstream Sentiment Valence Regression

We evaluate how our debiased word vectors make downstream tasks in sentiment analysis less discriminatory. There are many different ways to frame and measure sentiment in a sentence (i.e positive/negative, anger, sadness, valence). We focus on the task of regressing sentiment intensity or valence. [14] investigates the unfairness in this type of sentiment analysis task for over 200 different models trained on the SemEval-2018 Task 1 Affect in Tweets [18]. The authors create the *Equity Evaluation Corpus* (EEC) as a baseline to help measure differences in valence predictions between similar sentences that differ in the presence of a demographic identity term. The authors measure unfairness in valence regression for race using names that tend to belong to African American demographics vs European demographics. For example, a sentence template from the EEC dataset looks like *<Name> feels <emotional state word>*. For gender, the authors measure unfairness in valence regression using gendered words like he or she. Similar templates are used in this scenario (*<He/she> feels <emotional state word>*.) The authors perform valence predictions on sentences in the EEC database with emotional state words, and compare the average scores between different demographic groups. We describe the comparison metrics below.

- Avg. score difference **AA↑-↓E**: "The average Δ for only those pairs where the score for the African American noun phrase sentence is higher. The greater the magnitude of this score, the stronger the bias in systems that consistently give higher scores to African American-associated sentences."
- Avg. score difference **AA↓-↑E**: "The average Δ for only those pairs where the score for the African American noun phrase sentence is lower. The greater the magnitude of this score, the stronger the bias in systems that consistently give lower scores to African American-associated sentences."

The same metrics are used to compare Female (F) and Male (M) sentences (**F↑-↓M** and **F↓-↑M**).

We train a classical and deep regression model on the valence regression training set from SemEval-2018 Task 1 Affect in Tweets. As noted in [14] different model choices can result in varying degrees of bias. For example, a deeper model might be more sensitive to the subtle biases that enter either through the word embedding models or training set. Details for the two models are listed below.

- The classical model is the support vector regression algorithm (SVR). We encode text using the GloVe word embeddings trained on the Wikipedia corpus [22] and average the word vectors in the sentence. We feed the resulting vectors into the SVR model to predict a valance score between 0 and 1.
- Our deep model is an recurrent network with 128 LSTM units followed by a dense layer with 64 units. We use another single unit dense layer to output a valence prediction between 0 and 1. We use the mean squared error loss function. Finally, we encode text using the same GloVe word embeddings and represent a sentence as a padded matrix of word vectors which we pass into our model.

For each model, we train with the original GloVe word embeddings and GloVe word embeddings with debiased demographic identity terms. The resulting pearson correlations scores on the SemEval-2018 Task 1 gold standard set was 42%, 43% for the GloVe and Debiased GloVe SVR models respectively and 59% and 61% for the GloVe and Debiased GloVe LSTM models respectively. Using our word embeddings with debiased demographic attributes actually slightly improved the regression performance. This shows that our method does not adversely impact model performance when reducing word embedding sentiment bias.

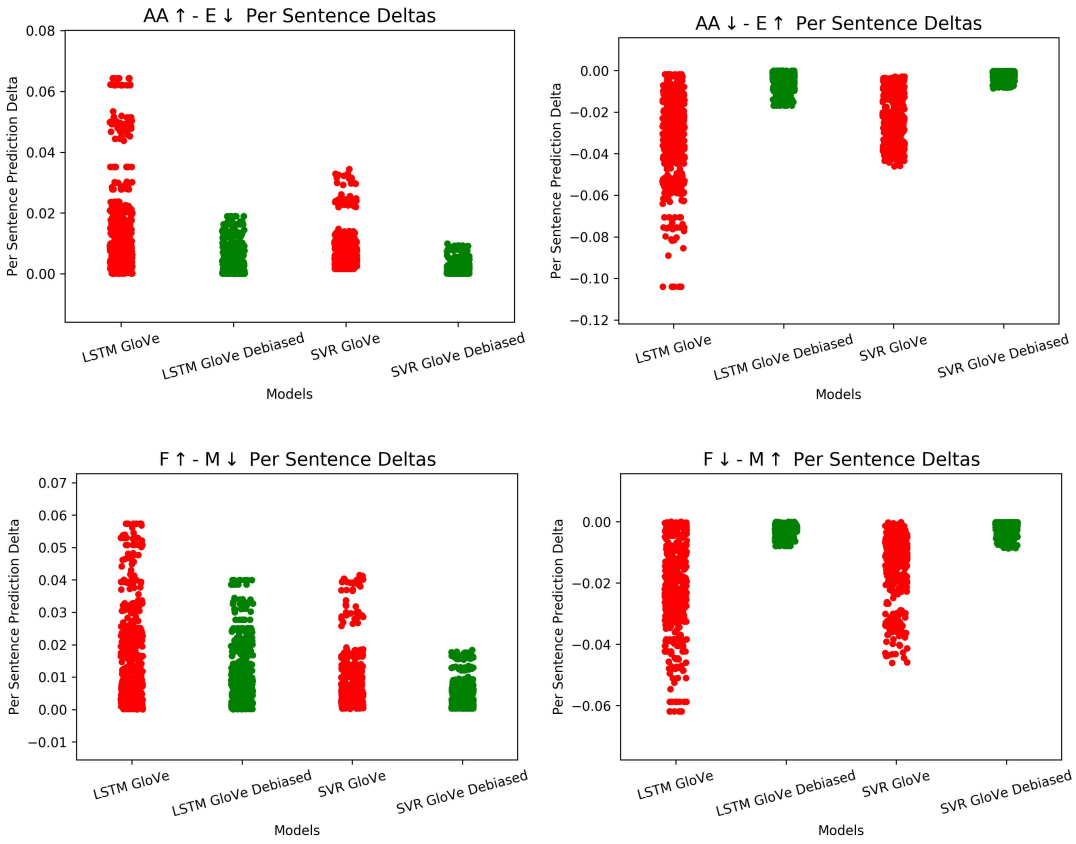

Figure 5: Scatter plots showing the distribution of sentiment valence regression score differences for four demographic group comparisons. For each group we measure score deltas for two types of models using our debiased word embeddings and original word embedding model. We measure score deltas on LSTM and Support Vector Regression algorithms to cover more complex and simpler models used in practice. In every category, our debiased word embedding models (shown in green) minimizes the bias between demographic groups with respect to valence predictions for similar sentences. Every model trained with our debiased word embeddings results in a distribution of scores with less variance and a closer mean to zero.

We also compare fairness for the four models via the avg. score differences for the groups: **AA↑-↓E**, **AA↓-↑E**, **F↑-↓M**, **F↓-↑M**, shown in Table 3. For these metrics, the smaller the magnitude, the less unfairness exists in the regression task. We see that for every category, our debiased word embeddings result in a smaller average gap between similar sentences with different demographic identity terms. To get a better sense for the distribution of pairwise valence score deltas for sentences with AA names vs E names and Male vs Female identity terms, we plotted a scatter plot in Figure 5. The dots are sentiment valence differences between the demographic groups for a particular sentence. Red dots are using the original word embedding models and green dots are using our debiased word embeddings. For every category, our algorithm compresses the variance in valence score deltas, showing that we have effectively reduced sentiment bias in a real world task. It is also interesting to note that the variance of scores for the SVR model is smaller than the LSTM model. This is likely due to the fact that deeper models can pick up on more subtle aspects of bias embedded in the word vector space. The remaining bias seen after using our word embeddings could come from other sources like unbalances in the training set or choice of learning algorithm. But we have shown that much of the sentiment bias for this task can be removed with our debiased word embeddings.

	Avg.Δ: AA↑-↓E	Avg.Δ: AA↓-↑E	Avg.Δ:F↑-↓M	Avg.Δ:F↓-↑E
LSTM GloVe	0.0159	0.0335	0.0166	0.0220
LSTM GloVe Debiased	0.0042	0.0046	0.0120	0.0024
Relative Bias Decrease	**73%**	**87%**	**28%**	**80%**
SVR GloVe	0.0085	0.0203	0.0098	0.0140
SVR GloVe Debiased	0.0021	0.0024	0.0054	0.0019
Relative Bias Decrease	**75%**	**90%**	**45%**	**86%**

Table 3: Table showing sentiment bias measures for 4 groups, on 4 different models. When training on our debiased word embeddings, we can achieve up to a 90 % decrease in bias via listed metrics.

4.5 Debiased Word Vectors Make Downstream Toxicity Classifier Fairer

We evaluate how we can use our method to create word vectors that are less correlated with toxicity. In many ways, toxicity can be thought of a subset of sentiment, where toxic concepts are related to more severe negative sentiment. On the other hand, the more positive the sentiment, the further a concept is from toxicity. Clearly this is not a perfect comparison, but we show that our debiasing technique for sentiment polarity can make toxicity prediction algorithms fairer. To adapt our debaising technique to focus on toxicity, one can try creating subspaces that represent *toxicity* and *non toxicity*, but this is a much more difficult task.

To show the stability of our adversarial framework, we also show how word embeddings that are depolarized with our technique make a downstream toxicity classifier fairer. [7] has made great strides on mitigating unfairness in toxicity classification, training a Convolution Neural Network (CNN) on a dataset of Wikipedia Talk Page comments that is balanced to mitigate dangerous over representations of certain demographic identity terms in toxic sentences. This helps the trained model make fewer false positive predictions for nontoxic sentences that contain a demographic identity term. In the paper, the authors propose a fairness metric called *Pinned AUC Equality Difference*, which measures area under ROC curve for a balanced test set of toxic and nontoxic synthetic sentences containing a particular identity term, t, from a set, T. The *Pinned AUC Equality Difference*, *pAUCed*, metric is presented below for convenience.

$$pAUCed = \sum_{t \in T} |AUC - pAUC_t|$$

Where AUC is the model's overall AUC and $pAUC_t$ is the AUC for the sentences containing the particular identity term, t.

Although rebalancing the training set helps the model make fewer dangerous false positive predictions, it misses much of the bias that stems from unfair toxicity correlations in the word embeddings. Figure 6 shows AUC results for 10 training runs of a CNN for partitions of the synthetic dataset containing each identity term. The top model is the CNN trained on the unbalanced Wikipedia Talk dataset. The second from the top is the results for the CNN trained on the balanced dataset via the method presented in [7]. Between these two graphs, one can see that there is less variance between the models performance on different identity terms, indicating less discriminatory behavior. We get less variance for the second from the bottom graph that shows the results for the model trained on the original and unbalanced Wikipedia Talk dataset with our detoxified identity term word vectors. Furthermore, we get even better results when combining the dataset debiasing technique with our word embeddings (bottom graph in Figure 6).

We now evaluate the toxicity classification models more formally. Table 4 shows the *Pinned AUC Equality Difference* metric for the 4 different types of model treatments. When just using our debiased word embeddings, we get a 52% increase in fairness via the *Pinned AUC Equality Difference* metric over the dataset debiasing technique developed in [7]. However, when combining this technique with our debiased word embeddings, we get the best results with a 59% percent increase in fairness via the *Pinned AUC Equality Difference* metric over the dataset debiasing treatment by itself. It is important to realize, that though our word embedding debiasing does a better job at creating a fairer model, it is mitigating a different source of bias. It is very possible that word embedding bias has a larger impact on model fairness than dataset bias. Still, we saw improved fairness when applying mitigation treatments at multiple levels of the NLP pipeline. Because unintended bias can enter a NLP pipeline at many stages, to achieve the fairest decision systems, we need to mitigate bias at multiple levels.

Finally, it is important to note that the toxicity classifier trained on our debiased embeddings not only minimized disparity in AUC distributions for various demographic identity terms, it also increased the average AUC for each group. In other words, the model more effectively learned how to distinguish toxic from non toxic statements, without using demographic information as a proxy. This is an encouraging sign that fairness need not always be at odds with accuracy.

	Pinned AUC Equality Difference
Original Model	5.900
Debiased Dataset Treatment	3.756
Debiased Word Embedding Treatment	1.768
Debiased Dataset and Word Embedding Treatment	1.534

Table 4: *Pinned AUC Equality Difference* metric for 4 different model debiasing treatments for toxicity classification. Using our debiased word embedding model we get a 70% increase in fairness compared to no debiased treatment, and a 52% improvement over the debiased dataset treatment baseline.

5 DISCUSSION

In this work, we discussed removing sentiment bias at the word embedding level. In NLP, word embeddings make up a substantial piece of machine learning pipelines. Consequently, word embeddings can have a large and adverse affect on the fairness of a downstream machine learning model. Mitigating sentiment bias and other related notions like toxicity bias at the word embedding level is an important step forward in creating fair machine learning models, especially in sentiment analysis systems. As [14] shows, there are many different sources of bias in machine learning systems (i.e embeddings, dataset, choice of learning algorithm, choice of thresholds). It is important to understand and mitigate bias at all stages of a ML pipeline to move forward. Our experiments in toxicity predictions showed that the best results with the smallest amount of harmful bias came from a combination of debiasing at the word embedding and dataset level.

In this work, we addressed sentiment bias with respect to the content of text, highlighting that demographic identity terms can contain large amounts of sentiment bias in word embedding models. Sentiment bias can also enter into text via more abstract concepts like phrases or even the stylistic choices of the author [9]. Mitigating other sources of bias are part of future work.

Our results showed that we can effectively decorrelate word vectors with sentiment. One of the insights from our attempts to debias word embeddings is that the vector space is fragile. Projecting word vectors to have no correlation with sentiment subspaces completely distorted the embeddings. Furthermore, any nonlinear

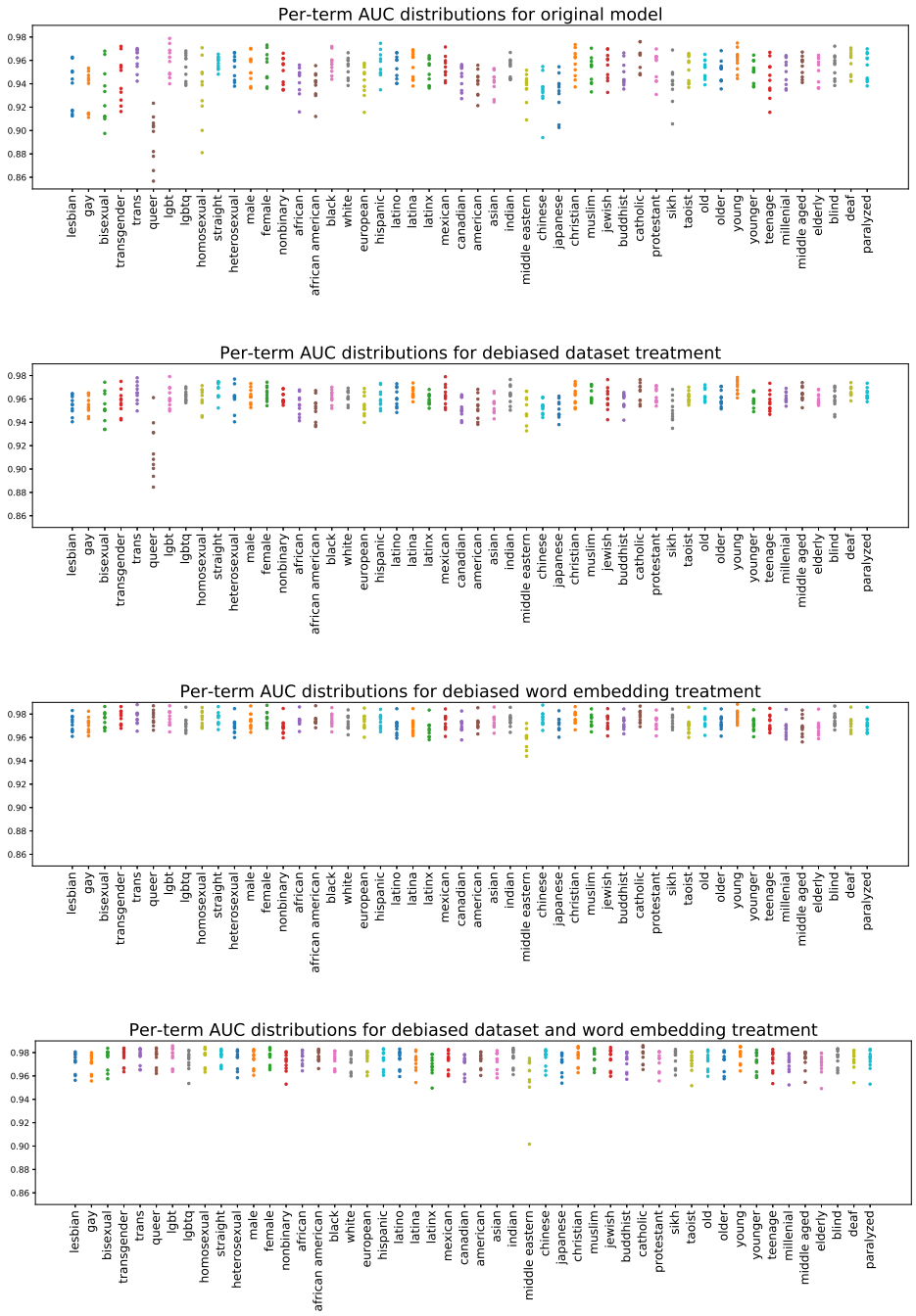

Figure 6: *Top*: Results for original toxicity classifier with no debiased treatment. *Second from the top*: Results for model trained on the debiased dataset technique from [7]. *Second from bottom*: Results for model trained with our debiased word embeddings. *Bottom*: Results for model trained with both the debiased dataset and debiased word embedding treatment. Graphs were generated using open source code: *https://github.com/conversationai/unintended-ml-bias-analysis*. Our technique, (*Second from bottom*) more effectively minimizes AUC discrepancies between demographic groups than the technique presented in [7]. Furthermore, combining our technique with the dataset debiasing technique presented in [7] yields the best results.

transformations to word vectors also proved to distort the word vectors' semantic meaning. We chose a simple linear model to be more sensitive to these distortions. Pairing the adversarial learning algorithm with our linear model enabled our method to minimize

correlation with sentiment, while retaining the usefulness of the word embedding for downstream NLP tasks. Furthermore, our linear model is interpretable, allowing one to understand how the adversarial learning algorithm is moving word vectors within the embedding space.

For our experiments, we also did not tailor our training set of word vectors for our adversarial algorithm to any group of demographic identity terms. One can imagine that having our model focus on depolarizing a subset of demographic attributes would yield better results for these word vectors. However, it is difficult to know ahead of time every possible demographic attribute one might want to debias. Consequently, we chose the route of choosing a large and general training set of words so our model can handle debiasing a larger set of demographic attributes.

Finally, in both our case studies (on sentiment valence regression and toxicity classification) we noticed that the debiased word embeddings made improvements not only in the fairness metrics proposed in [14] and [7], but also in the accuracy of the overall classifier. This is an important notion, as unfair biases in the data could be causing a machine learning model to overfit to demographic information not relevant to the problem tasked to solve. For example, biased word embeddings learn to unfairly associate demographic identity terms with negative or positive sentiment due to biases in word corpora. A machine learning model trained to distinguish between positive and negative sentiment can overfit to the biased initialization from the word vectors, causing it to perform worse for the general case. Preventing the machine learning model from picking up on unfairly biased word embedding initialization goes a long way in enabling the model to focus on more general and correct signals for predicting positive or negative sentiment.

6 CONCLUSION

Word embeddings are the bedrock for many applications in NLP. Unfair sentiment polarity between different demographic identity terms can cause discrimination in downstream applications. Unfortunately unlike gender bias, which has been thoroughly explored in the research community, sentiment bias is a fairly loose concept. A systematic solution for this problem is imperative to mitigate the risk of discrimination in NLP models. In this work, we proposed an adversarial training algorithm to reduce word vector sentiment bias for demographic identity terms. We are able to reduce sentiment polarity for many types of demographic attributes in text such as religious identifiers, names, and gendered terms, without distorting the embedding model. Furthermore, when using our debiased word embeddings in realistic downstream sentiment regression and toxicity prediction tasks, we showed large increases in fairness via defined metrics.

7 ACKNOWLEDGMENTS

This work was made possible in part through support of the United States Agency for International Development. The opinions expressed herein are those of the authors and do not necessarily reflect the views of the United States Agency for International Development or the US Government.

REFERENCES

[1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL*. 19–27.
[2] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *FATML* (2017).
[3] Su Lin Blodgett and Brendan O'Connor. 2017. Racial Disparity in Natural Language Processing: A Case Study of Social Media African-American English. *FATML* (2017).
[4] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Quantifying and reducing stereotypes in word embeddings. *ICML* (2016).
[5] 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 *NIPS*. 4349–4357.
[6] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.
[7] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. *AAAI* (2018).
[8] Harrison Edwards and Amos Storkey. 2016. Censoring representations with an adversary. *ICLR* (2016).
[9] Yanai Elazar and Yoav Goldberg. 2018. Adversarial Removal of Demographic Attributes from Text Data. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018).
[10] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences* 115, 16 (2018).
[11] Hila Gonen and Yoav Goldberg. 2019. Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them. *NAACL* (2019). arXiv:1903.03862
[12] Dirk Hovy and Shannon L. Spruit. 2016. The Social Impact of Natural Language Processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 591–598. https://doi.org/10.18653/v1/P16-2096
[13] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *ACM*. 168–177.
[14] Svetlana Kiritchenko and Saif M Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. *Proceedings of the 7thJoint Conference on Lexical and Computational Se-mantics(*SEM), New Orleans, USA*. (2018).
[15] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.
[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.
[18] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 Task 1: Affect in Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 1–17. https://doi.org/10.18653/v1/S18-1001
[19] Ben Packer, Yoni Halpern, Mario Guajardo-CÃĪspedes, and Margaret Mitchell. 2018. Text Embedding Models Contain Bias. Here's Why That Matters. Google Developers.
[20] Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *FTIR* 2, 1–2 (2008), 1–135.
[21] Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing Gender Bias in Abusive Language Detection. *EMNLP* (2018).
[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.
[23] Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. 2018. Fairness GAN. *arXiv preprint arXiv:1805.09910* (2018).
[24] Chris Sweeney and Maryam Najafian. 2019. A Transparent Framework for Evaluating Unintended Demographic Bias in Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1662–1667. https://doi.org/10.18653/v1/P19-1162
[25] Rachael Tatman. 2017. Gender and Dialect Bias in YouTube's Automatic Captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. 53–59.
[26] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP*. 347–354.
[27] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World*

Wide Web. International World Wide Web Conferences Steering Committee, 1391–1399.

[28] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. *AIES* (2018).