Attenuating Bias in Word Vectors

Sunipa Dev
University of Utah

Jeff M Phillips
University of Utah

Abstract

Word vector representations are well developed tools for various NLP and Machine Learning tasks and are known to retain significant semantic and syntactic structure of languages. But they are prone to carrying and amplifying bias which can perpetrate discrimination in various applications. In this work, we explore new simple ways to detect the most stereotypically gendered words in an embedding and remove the bias from them. We verify how names are masked carriers of gender bias and then use that as a tool to attenuate bias in embeddings. Further, we extend this property of names to show how names can be used to detect other types of bias in the embeddings such as bias based on race, ethnicity, and age.

1 BIAS IN WORD VECTORS

Word embeddings are an increasingly popular application of neural networks wherein enormous text corpora are taken as input and words therein are mapped to a vector in some high dimensional space. Two commonly used approaches to implement this are WordToVec [16, 15] and GloVe [17]. These word vector representations estimate similarity between words based on the context of their nearby text, or to predict the likelihood of seeing words in the context of another. Richer properties were discovered such as synonym similarity, linear word relationships, and analogies such as `man` : `woman` :: `king` : `queen`. Their use is now standard in training complex language models.

However, it has been observed that word embeddings are prone to express the bias inherent in the data it is extracted from [2, 3, 6]. Further, Zhao *et al.* (2017) [18] and Hendricks *et al.* (2018) [5] show that machine learning algorithms and their output show more bias than the data they are generated from.

Word vector embeddings as used in machine learning towards applications which significantly affect people's lives, such as to assess credit [11], predict crime [4], and other emerging domains such judging loan applications and resumes for jobs or college applications. So it is paramount that efforts are made to identify and if possible to remove bias inherent in them. Or at least, we should attempt to minimize the propagation of bias within them. For instance, in using existing word embeddings, Bolukbasi *et al.* (2016) [2] demonstrated that women and men are associated with different professions, with men associated with leaderships roles and professions like doctor, programmer and women closer to professions like receptionist or nurse. Caliskan *et al.* (2017) [6] similarly noted how word embeddings show that women are more closely associated with arts than math while it is the opposite for men. They also showed how positive and negative connotations are associated with European-American versus African-American names.

Our work simplifies, quantifies, and fine-tunes these approaches: we show that very simple linear projection of all words based on vectors captured by common names is an effective and general way to significantly reduce bias in word embeddings. More specifically:

1a. We demonstrate that simple linear projection of all word vectors along a bias direction.

1b. We show that these results can be slightly improved by dampening the projection of words

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

Acknowledgements : Thanks to NSF CCF-1350888, ACI-1443046, CNS-1514520, CNS-1564287, IIS-1816149 and NVidia Corporation. Part of the work by JP was done while visiting the Simons Institute for Theory of Computing.

In this paper, the use of the term "bias" is meant in the statistical sense, as a deviation from a population parameter, not implying intent.

which are far from the projection distance. Further, simple linear projection is more effective than the Hard Debiasing of Bolukbasi *et al.* (2016) [2] which is more complex and also partially relies on crowd sourcing.

2. We examine the bias inherent in the standard word pairs used for debiasing based on gender by randomly flipping or swapping these words in the raw text before creating the embeddings. We show that this alone does not eliminate bias in word embeddings, corroborating that simple language modification is not as effective as repairing the word embeddings themselves.

3a. We show that common names with gender association (e.g., `john`, `amy`) often provides a more effective gender subspace to debias along than using gendered words (e.g., `he`, `she`).

3b. We demonstrate that names carry other inherent, and sometimes unfavorable, biases associated with race, nationality, and age, which also corresponds with bias subspaces in word embeddings. And that it is effective to use common names to establish these bias directions and remove this bias from word embeddings.

4. We also propose and demonstrate the use of two quantitative tests for evaluating how gender biased an embedding is.

2 DATA AND NOTATIONS

We set as default the text corpus of a English Wikipedia dump (`dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2`) with 4.57 billion tokens and we extract a GloVe embedding from it in $D = 300$ dimensions per word. We restrict the word vocabulary to the most frequent 100,000 words. We also modify the text corpus and extract embeddings from it as described later. So, for each word in the Vocabulary W, we represent the word by the vector $w_i \in \mathbb{R}^D$ in the embedding. The bias (e.g., gender) subspace is denoted by a set of vector B. It is typically considered in this work to be a single unit vector, v_B (explained in detail later). As we will revisit, a single vector is typically sufficient, and will simplify descriptions. However, these approaches can be generalized to a set of vectors defining a multi-dimensional subspace.

3 HOW TO ATTENUATE BIAS

Given a word embedding, debiasing typically takes as input a set $\mathcal{E} = \{E_1, E_2, \ldots, E_m\}$ of equality sets. An equality set E_j for instance can be a single pair (e.g., {`man`, `woman`}), but could be more words (e.g., {`latina`, `latino`, `latinx`}) that if the bias connotation (e.g, gender) is removed, then it would objectively make sense for all of them to be equal. Our data sets will only use word pairs (as a default the ones in Table 1), and we will describe them as such hereafter for simpler descriptions. In particular, we will represent each E_j as a set of two vectors $e_i^+, e_i^- \in \mathbb{R}^D$.

{`man`,`woman`}, {`son`,`daughter`}, {`he`,`she`}, {`his`,`her`}, {`male`,`female`}, {`boy`,`girl`}, {`himself`,`herself`}, {`guy`,`gal`}, {`father`,`mother`}, {`john`,`mary`}

Table 1: Gendered Word Pairs

Given such a set $\mathcal{E}$ of equality sets, the bias vector v_B can be formed as follows [2]. For each $E_j = \{e_j^+, e_j^-\}$ create a vector $\vec{e}_i = e_i^+ - e_i^-$ between the pairs. Stack these to form a matrix $Q = [\vec{e}_1\ \vec{e}_2\ \ldots\ \vec{e}_m]$, and let v_B be the top singular vector of Q. We revisit how to create such a bias direction in Section 4.

Now given a word vector $w \in W$, we can project it to its component along this bias direction v_B as

$$\pi_B(w) = \langle w, v_B \rangle v_B.$$

3.1 Existing Method : Hard Debiasing

The most notable advance towards debiasing embeddings along the gender direction has been by Bolukbasi *et al.* (2016) [2] in their algorithm called Hard Debiasing (HD). It takes a set of words desired to be neutralized, $\{w_1, w_2, \ldots, w_n\} = W_N \subset W$, a unit bias subspace vector v_B, and a set of equality sets $E_1, E_2, \ldots, E_m$.

First, words $\{w_1, w_2, \ldots, w_n\} \in W_N$ are projected orthogonal to the bias direction and normalized

$$w_i' = \frac{w_i - w_B}{||w_i - w_B||}.$$

Second, it corrects the locations of the vectors in the equality sets. Let $\mu_j = \frac{1}{|E|}\sum_{e \in E_j} e$ be the mean of an equality set, and $\mu = \frac{1}{m}\sum_{j=1}^{m} \mu_j$ be the mean of of equality set means. Let $\nu_j = \mu - \mu_j$ be the offset of a particular equality set from the mean. Now each $e \in E_j$ in each equality set E_j is first centered using their average and then neutralized as

$$e' = \nu_j + \sqrt{1 - \|\nu_j\|^2}\frac{\pi_B(e) - v_B}{\|\pi_B(e) - v_B\|}.$$

Intuitively ν_j quantifies the amount words in each equality set E_j differ from each other in directions apart from the gender direction. This is used to center the words in each of these sets.

This renders word pairs such as `man` and `woman` as equidistant from the neutral words w_i' with each word

of the pair being centralized and moved to a position opposite the other in the space. This can filter out properties either word gained by being used in some other context, like `mankind` or `humans` for the word `man`.

The word set $W_N = \{w_1, w_2, \ldots, w_n\} \subset W$ which is debiased is obtained in two steps. First it seeds some words as definitionally gendered via crowd sourcing and using dictionary definitions; the complement – ones not selected in this step – are set as neutral. Next, using this seeding an SVM is trained and used to predict among all W the set of other biased W_B or neutral words W_N. This set W_N is taken as desired to be neutral and is debiased. Thus not all words W in the vocabulary are debiased in this procedure, only a select set chosen via crowd-sourcing and definitions, and its extrapolation. Also the word vectors in the equality sets are also handled separately. This makes this approach not a fully automatic way to debias the vector embedding.

3.2 Alternate and Simple Methods

We next present some simple alternatives to HD which are simple and fully automatic. These all assume a bias direction v_B.

Subtraction. As a simple baseline, for *all* word vectors w subtract the gender direction v_B from w:

$$w' = w - v_B.$$

Linear Projection. A better baseline is to project *all* words $w \in W$ orthogonally to the bias vector v_B.

$$w' = w - \pi_B(w) = w - \langle w, v_B \rangle v_B.$$

This enforces that the updated set $W' = \{w' \mid w \in W\}$ has no component along v_B, and hence the resulting span is only $D - 1$ dimensions. Reducing the total dimension from say 300 to 299 should have minimal effects of expressiveness or generalizability of the word vector embeddings.

Bolukbasi *et al.* [2] apply this same step to a dictionary definition based extrapolation and crowd-source-chosen set of word pairs $W_N \subset W$. We quantify in Section 5 that this single universal projection step debiases better than HD.

For example, consider the bias as gender, and the equality set with words `man` and `woman`. Linear projection will subtract from their word embeddings the proportion that were along the gender direction v_B learned from a larger set of equality pairs. It will make them close-by but not exactly equal. The word `man` is used in many extra senses than the word `woman`; it is used to refer to humankind, to a person in general,

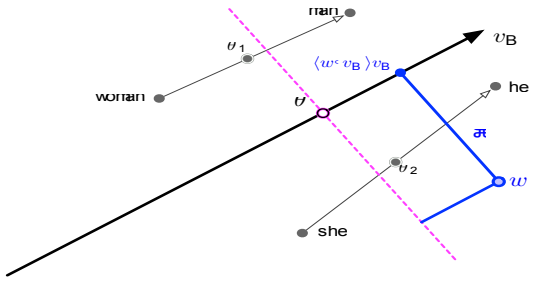

Figure 1: Illustration of η and β for word vector w.

and in expressions like "oh man". In contrast a simpler word pair with fewer word senses, like (`he - she`) and (`him - her`), we can expect them to be almost at identical positions in the vector space after debiasing, implying their synonymity.

Thus, this approach uniformly reduces the component of the word along the bias direction without compromising on the differences that words (and word pairs) have.

3.3 Partial Projection

A potential issue with the simple approaches is that they can significantly change some embedded words which are definitionally biased (e.g., the neutral words W_B described by Bolukbasi *et al.* [2]). *[[We note that this may not *actually* be a problem (see Section 5); the change may only be associated with the bias, so removing it would then not change the meaning of those words in any way except the ones we want to avoid.]]* However, these intuitively should be words which have correlation with the bias vector, but also are far in the orthogonal direction. In this section we explore how to automatically attenuate the effect of the projection on these words.

This stems from the observation that given a bias direction, the words which are most extreme in this direction (have the largest dot product) sometimes have a reasonable biased context, but some do not. These "false positives" may be large normed vectors which also happen to have a component in the bias direction.

We start with a bias direction v_B and mean μ derived from equality pairs (defined the same way as in context of HD). Now given a word vector w we decompose it into key values along two components, illustrated in Figure 1. First, we write its bias component as

$$\beta(w) = \langle w, v_B \rangle - \langle \mu, v_B \rangle.$$

This is the difference of w from μ when both are projected onto the bias direction v_B.

Second, we write a (residual) orthogonal component

$$r(w) = w - \langle w, v_B \rangle v_B.$$

Let $\eta(w) = \|r(w)\|$ be its value. It is the orthogonal distance from the bias vector v_B; recall we chose v_B to pass through the origin, so the choice of μ does not affect this distance.

Now we will maintain the orthogonal component ($r(w)$, which is in a subspace spanned by $D-1$ out of D dimensions) but adjust the bias component $\beta(w)$ to make it closer to μ. But the adjustment will depend on the magnitude $\eta(w)$. As a default we set

$$w' = \mu + r(w)$$

so all word vectors retain their orthogonal component, but have a fixed and constant bias term. This is functionally equivalent to the Linear Projection approach; the only difference is that instead of having a 0 magnitude along v_B (and the orthogonal part unchanged), it instead has a magnitude of constant μ along v_B (and the orthogonal part still unchanged). This adds a constant to every inner product, and a constant offset to any linear projection or classifier. If we are required to work with normalized vectors (we do not recommend this as the vector length captures veracity information about its embedding), we can simple set $w' = r(w)/\|r(w)\|$.

Given this set-up, we now propose three modifications. In each set

$$w' = \mu + r(w) + \beta \cdot f_i(\eta(w)) \cdot v_B$$

were f_i for $i = \{1, 2, 3\}$ is a function of only the orthogonal value $\eta(w)$. For the default case $f(\eta) = 0$

$$f_1(\eta) = \sigma^2/(\eta + 1)^2$$
$$f_2(\eta) = \exp(-\eta^2/\sigma^2)$$
$$f_3(\eta) = \max(0, \sigma/2\eta)$$

Here σ is a hyperparameter that controls the importance of η; in the Appendix **??** we show that we can just set $\sigma = 1$.

In Figure 2 we see the regions of the (η, β)-space that the functions f, f_1 and f_2 consider gendered. f projects all points onto the $y = \mu$ line. But variants f_1, f_2, and f_3 are represented by curves that dampen the bias reduction to different degrees as η increases. Points P1 and P2 have the same dot products with the bias direction but different dot products along the other $D-1$ dimensions. We can observe the effects of each dampening function as η increases from P1 to P2.

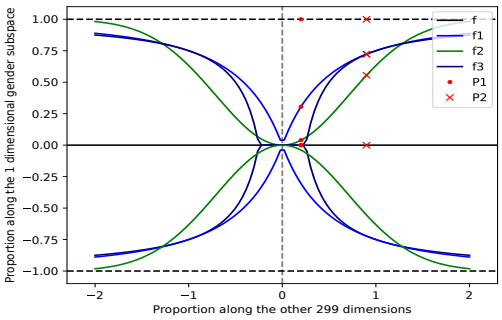

Figure 2: The gendered region as per the three variations of projection. Both points P1 and P2 have a dot product of 1.0 initially with the gender subspace. But their orthogonal distance to it differs, as expressed by their dot product with the other 299 dimensions.

3.4 Flipping the Raw Text

Since the embeddings preserve inner products of the data from which it is drawn, we explore if we can make the data itself gender unbiased and then observe how that change shows up in the embedding. Unbiasing a textual corpus completely can be very intricate and complicated since there are a many (sometimes implicit) gender indicators in text. Nonetheless, we propose a simple way of neutralizing bias in textual data by using word pairs $E_1, E_2, \ldots E_m$; in particular, when we observe in raw text on part of a word part, we randomly flip it to the other pair. For instance for gendered word pairs (e.g., (`he` - `she`)) in a string "`he` was a doctor" we may flip to "`she` was a doctor."

We implement this procedure over the entire input raw text, and try various probabilities of flipping each observed word, focusing on probabilities 0.5, 0.75 and 1.00. The first 0.5-flip probability makes each element of a word pair equally likely. The last 1.00-flip probability reverses the roles of those word pairs, and 0.75-flip probability does something in between. We perform this set of experiments on the default Wikipedia data set and switch between word pairs (say `man` → `woman`, `she` → `he`, etc), from a list larger that Table 2 consisting of 75 worxd pairs; (see full version [7]).

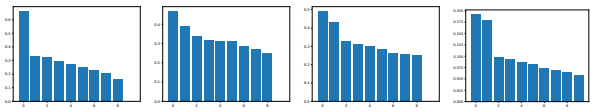

Figure 3: Fractional singular values for avg male - female words (as per Table 1) after flipping with probability (from left to right) 0.0 (the original data set), 0.5, 0.75, and 1.0.

Table 2: Some of the most gendered words in default embedding; and most gendered adjectives and occupation words.

Gendered Words			
miss	herself	forefather	himself
maid	heroine	nephew	congressman
motherhood	jessica	zahir	suceeded
adriana	seductive	him	sir

Female Adjectives	Male Adjectives
glamorous	strong
diva	muscular
shimmery	powerful
beautiful	fast
Female Occupations	**Male Occupations**
nurse	soldier
maid	captain
housewife	officer
prostitute	footballer

We observe how the proportion along the principal component changes with this flipping in Figure 3. We see that flipping with 0.5 somewhat dampens the difference between the different principal components. On the other hand flipping with probability 1.0 (and to a lesser extent 0.75) exacerbates the gender components rather than dampening it. Now there are two components significantly larger than the others. This indicates this flipping is only addressing part of the explicit bias, but missing some implicit bias, and these effects are now muddled.

4 THE BIAS SUBSPACE

We explore ways of detecting and defining the bias subspace v_B and recovering the most gendered words in the embedding. Recall as default, we use v_B as the top singular vector of the matrix defined by stacking vectors $\vec{e_i} = e_i^+ - e_i^-$ of biased word pairs. We primarily focus on gendered bias, using words in Table 1, and show later how to effectively extend to other biases.

Most gendered words. The dot product, $\langle v_B, w \rangle$ of the word vectors w with the gender subspace v_B is a good indicator of how gendered a word is. The magnitude of the dot product tells us of the length along the gender subspace and the sign tells us whether it is more female or male. Some of the words denoted as most gendered are listed in Table 2.

4.1 Bias Direction using Names

When listing gendered words by $|\langle v_B, w \rangle|$, we observe that many gendered words are names. This indicates the potential to use names as an alternative (and potentially in a more general way) to bootstrap finding the gender direction.

From the top 100K words, we extract the 10 most common male $\{m_1, m_2, \ldots, m_{10}\}$ and female $\{s_1, s_2, \ldots, s_{10}\}$ names which are not used in ambiguous ways (e.g., not the name `hope` which could also refer to the sentiment). We pair these 10 names from each category (male, female) randomly and compute the SVD as before. We observe in Supplmentary Material **??** that the fractional singular values show a similar pattern as with the list of correctly gendered word pairs like (`man - woman`), (`he - she`), etc.

But this way of pairing names is quite imprecise. These names are not 'opposites' of each other in the sense that word pairs are. So, we modify and propose a **2-means** method on how to compute v_B so as to better use names to detect the bias in the embedding. This method gives us this advantage where we do not necessarily need word pairs or equality sets as in Bolukbasi *et al.* [2].

By the **2-means** method, our gender direction is calculated as,

$$v_{B,\text{names}} = \frac{s - m}{\|s - m\|},$$

where $s = \frac{1}{10}\sum_i s_i$ and $m = \frac{1}{10}\sum_i m_i$.

Using the default Wikipedia dataset, we found that this is a good approximator of the gender subspace defined by the first right singular vector calculated using gendered words from Table 1; there dot product is 0.809. We find similar large dot product scores for other datasets too. There too we collect all the most gendered words as per the gender direction $v_{B,\text{names}}$ determined by these names. Most gendered words returned are similar as using the default v_B, like occupational words, adjectives, and synonyms for each gender. We find names to express similar classification of words along male - female vectors with `homemaker` more female and `policeman` being more male. We illustrate this in more detail in the full version [7].

5 QUANTIFYING BIAS

In this section we develop new measures to quantify how much bias has been removed from an embedding, and evaluate the various techniques we have developed for doing so.

As one measure, we use the Word Embedding Association Test (WEAT) test developed by Caliskan *et al.* (2017) [6] as analogous to the IAT tests to evaluate the association of male and female gendered words with two categories of target words: career oriented words versus family oriented words. We detail WEAT and list the exact words used (as in [6]) in the full version [7]; smaller values are better.

Bolukbasi *et al.* [2] evaluated embedding bias use a crowdsourced judgement of whether an analogy produced by an embedding is biased or not. Our goal was

to avoid crowd sourcing, so we propose two more automatic tests to qualitatively and uniformly evaluate an embedding for the presence of gender bias.

Embedding Coherence Test (ECT). A way to evaluate how the neutralization technique affects the embedding is to evaluate how the nearest neighbors change for (a) gendered pairs of words $\mathcal{E}$ and (b) indirect-bias-affected words such as those associated with sports or occupational words (e.g., `football`, `captain`, `doctor`). We use the gendered word pairs in Table 1 for $\mathcal{E}$ and the professions list $P = \{p_1, p_2, \ldots, p_k\}$ as proposed and used by Bolukbasi *et al.* `https://github.com/tolga-b/debiaswe` (see also word lists in full version [7]) to represent (b).

S1: For all word pair $\{e_j^+, e_j^-\} = E_j \in \mathcal{E}$ we compute two means $m = \frac{1}{|\mathcal{E}|}\sum_{E_j \in \mathcal{E}} e_j^+$ and $s = \frac{1}{|\mathcal{E}|}\sum_{E_j \in \mathcal{E}} e_j^-$. We find the cosine similarity of both m and s to all words $p_i \in P$. This creates two vectors $u_m, u_s \in \mathbb{R}^k$.

S2: We transform these similarity vectors to replace each coordinate by its rank order, and compute the Spearman Coefficient (in $[-1, 1]$, larger is better) between the rank order of the similarities to words in P.

Thus, here, we care about the order in which the words in P occur as neighbors to each word pair rather than the exact distance. The exact distance between each word pair would depend on the usage of each word and thus on all the different dimensions other than the gender subspace too. But the order being relatively the same, as determined using Spearman Coefficient would indicate the dampening of bias in the gender direction (i.e., if `doctor` by profession is the 2nd closest of all professions to both `man` and `woman`, then the embedding has a dampened bias for the word `doctor` in the gender direction). Neutralization should ideally bring the Spearman coefficient towards 1.

Embedding Quality Test (EQT). The demonstration by Bolukbasi *et al.* [2] about the skewed gender roles in embeddings using analogies is what we try to quantify in this test. We attempt to quantify the improvement in analogies with respect to bias in the embeddings. We use the same sets $\mathcal{E}$ and P as in the ECT test. However, for each profession $p_i \in P$ we create a list S_i of their plurals and synonyms from WordNet on NLTK [14].

S1: For each word pair $\{e_j^+, e_j^-\} = E_j \in \mathcal{E}$, and each occupation word $p_i \in P$, we test if the analogy $e_j^+ : e_j^- :: p_i$ returns a word from S_i. If yes, we set $Q(E_j, p_i) = 1$, and $Q(E_j, p_i) = 0$ otherwise.

S2: Return the average value across all combinations $\frac{1}{|\mathcal{E}|}\frac{1}{k}\sum_{E_j \in \mathcal{E}}\sum_{p_i \in P} Q(E_j, p_i)$.

The scores for EQT are typically much smaller than for ECT. We explain two reasons for this.

First, EQT does not check for if the analogy makes relative sense, biased or otherwise. So, "`man` : `woman` :: `doctor` : `nurse`" is as wrong as "`man` : `woman` :: `doctor` : `chair`." This pushes the score down.

Second, synonyms in each set s_i as returned by WordNet [8] on the Natural Language Toolkit, NLTK [14] do not always contain all possible variants of the word. For example, the words `psychiatrist` and `psychologist` can be seen as analogous for our purposes here but linguistically are removed enough that WordNet does not put them as synonyms together. Hence, even after debiasing, if the analogy returns "`man` : `woman` :: `psychiatrist` : `psychologist`" S1 returns 0. Further, since the data also has several misspelt words, `archeologist` is not recognized as a synonym or alternative for the word `archaeologist`. For this too S1 returns a 0.

The first caveat can be side-stepped by restricting the pool of words we search over for the analogous word to be from list P. But it is debatable if an embedding should be penalized equally for returning both nurse or chair for the analogy "`man` : `woman` :: `doctor` : ?"

This measures the quality of analogies, with better quality having a score closer to 1.

Evaluating embeddings. We mainly run 4 methods to evaluate our methods WEAT, EQT, and two variants of ECT: ECT (word pairs) uses $\mathcal{E}$ defined by words in Table 1 and ECT (names) which uses vectors m and s derived by gendered names.

We observe in Table 4 that the ECT score increases for all methods in comparison to the non-debiased (the original) word embedding; the exception is flipping with 1.0 probability score for ECT (word pairs) and all flipping variants for ECT (names). Flipping does nothing to affect the names, so it is not surprising that it does not improve this score; further indicating that it is challenging to directly fix bias in raw text before creating embeddings. Moreover, HD has the lowest score (of 0.917) whereas projection obtains scores of 0.996 (with v_B) and 0.943 (with $v_{B,\mathsf{names}}$).

EQT is a more challenging test, and the original embedding only achieves a score of 0.128, and HD only obtains 0.145 (that is $12 - 15\%$ of occupation words have their related word as nearest neighbor). On the other hand, projection increases this percentage to 28.3% (using v_B) and 29.1% (using $v_{B,\mathsf{names}}$). Even subtraction does nearly as well at between $23 - 27\%$.

Table 3: What analogies look like before and after damping gender by different methods discussed : hard debiasing (HD), flipping words in text corpus, subtraction and projection

analogy head	original	HD	flipping			subtraction	projection
			0.5	0.75	1.0		
man : woman :: doctor :	nurse	surgeon	dr	dr	medicine	physician	physician
man : woman :: footballer :	politician	striker	midfielder	goalkeeper	striker	politician	midfielder
he : she :: strong :	weak	stronger	weak	strongly	many	well	stronger
he : she :: captain :	mrs	lieutenant	lieutenant	colonel	colonel	lieutenant	lieutenant
john : mary :: doctor :	nurse	physician	medicine	surgeon	nurse	father	physician

Table 4: Performance on ECT, EQT and WEAT by the different debiasing methods; and performance on standard similarity and analogy tests.

analogy head	original	HD	flipping			subtraction		projection	
			0.5	0.75	1.0	word pairs	names	word pairs	names
ECT (word pairs)	0.798	0.917	0.983	0.984	0.683	0.963	0.936	0.996	0.943
ECT (names)	0.832	0.968	0.714	0.662	0.587	0.923	0.966	0.935	0.999
EQT	0.128	0.145	0.131	0.098	0.085	0.268	0.236	0.283	0.291
WEAT	1.623	1.221	1.164	1.09	1.03	1.427	1.440	1.233	1.219
WSim	0.637	0.537	0.567	0.537	0.536	0.627	0.636	0.627	0.629
Simlex	0.324	0.314	0.317	0.314	0.264	0.302	0.312	0.321	0.321
Google Analogy	0.623	0.561	0.565	0.561	0.321	0.538	0.565	0.565	0.584

Generally, the subtraction always performs slightly worse than projection.

For the WEAT test, the original data has a score of 1.623, and this is decreased the most by all forms of flipping, down to about 1.1. HD and projection do about the same with HD obtaining a score of 1.221 and projection obtaining 1.219 (with $v_{B,\mathsf{names}}$) and 1.234 (with v_B); values closer to 0 are better (See full version for details of WEAT). In the bottom of Table 4 we also run these approaches on standard similarity and analogy tests for evaluating the quality of embeddings. We use cosine similarity [13] on WordSimilarity-353 (WSim, 353 word pairs) [9] and SimLex-999 (Simlex, 999 word pairs) [10], each of which evaluates a Spearman coefficient (larger is better). We also use the Google Analogy Dataset using the function 3COSADD [12] which takes in three words which for a part of the analogy and returns the 4th word which fits the analogy the best.

We observe (as expected) that all that debiasing approaches reduce these scores. The largest decrease in scores (between 1% and 10%) is almost always from HD. Flipping at 0.5 rate is comparable to HD. And simple linear projection decreases the least (usually only about 1%, except on analogies where it is 7% (with v_B) or 5% (with $v_{B,\mathsf{names}}$).

In Table 5 we also evaluate the damping mechanisms defined by f_1, f_2, and f_3, using v_B. These are very comparable to simple linear projection (represented by f). The scores for ECT, EQT, and WEAT are all about the same as simple linear projection, usually slightly worse.

Table 5: Performance of damped linear projection using word pairs.

Tests	f	f_1	f_2	f_3
ECT	0.996	0.994	0.995	0.997
EQT	0.283	0.280	0.292	0.287
WEAT	1.233	1.253	1.245	1.241
WSim	0.627	0.628	0.627	0.627
Simlex	0.321	0.324	0.324	0.324
Google Analogy	0.565	0.571	0.569	0.569

While ECT, EQT and WEAT scores are in a similar range for all of f, f_1, f_2, and f_3; the dampened approaches f_1, f_2, and f_3 performs better on the Google Analogy test. This test set is devoid of bias and is made up of syntactic and semantic analogies. So, a score closer to that of the original, biased embedding, tells us that more structure has been retained by f_1, f_2 and f_3. Overall, any of these approaches could be used if a user wants to debias while retaining as much structure as possible, but otherwise linear projection (or f) is roughly as good as these dampened approaches.

6 DETECTING OTHER BIAS USING NAMES

We saw so far how projection combined with finding the gender direction using names works well and works as well as projection combined with finding the gender direction using word pairs. We explore here a way of extending this approach to detect other kinds of bias where we cannot necessarily find good word pairs to indicate a direction, like Table 1 for gender, but where names are known to belong to certain protected demographic groups. For example, there is a di-

vide between names that different racial groups tend to use more. Caliskan *et al.* [6] use a list of names that are more African-American (AA) versus names that are more European-American (EA) for their analysis of bias. There are similar lists of names that are distinctly and commonly used by different ethnic, racial (e.g., Asian, African-American) and even religious (for e.g., Islamic) groups. We first try this with two common demographic group divides : Hispanic / European-American and African-American / European-American.

Hispanic and European-American names. Even though we begin with most commonly used Hispanic (H) names (word lists in full version [7]), this is tricky as not all names occur as much as European American names and are thus not as well embedded. We use the frequencies from the dataset to guide us in selecting commonly used names that are also most frequent in the Wikipedia dataset. Using the same method as Section 4.1, we determine the direction, $v_{B,\text{names}}$, which encodes this racial difference and find the words most commonly aligned with it. Other Hispanic and European-American names are the closest words. But other words like, `latino` or `hispanic` also appear to be close, which affirms that we are capturing the right subspace.

African-American and European-American names. We see a similar trend when we use African-American names and European-American names (Figure 4). We use the African-American names used by Caliskan *et al.* (2017) [6]. We determine the bias direction by using method in Section 4.1.

We plot in Figure 4 a few occupation words along the axes defined by H-EA and AA-EA bias directions, and compare them with those along the male-female axis. The embedding is different among the groups, and likely still generally more subordinate-biased towards Hispanic and African-American names as it was for female. Although `footballer` is more Hispanic than European-American, while `maid` is more neutral in the racial bias setting than the gender setting. We see this pattern repeated across embeddings and datasets (see Appendix **??** of full version [7]).

When we switch the type of bias, we find different patterns in the embeddings. In the case of both of these racial directions, there is a the split in not just occupation words but other words that are detected as highly associated with the bias subspace. It shows up foremost among the closest words of the subspace of the bias. Here, we find words like `drugs` and `illegal` close to the H-EA direction while, close to the AA-EA direction, we retrieve several slang words used to refer to African-Americans. These word associations with each racial group can be detected by the WEAT tests (lower means less bias) using positive and negative words as demonstrated by Caliskan *et al.* (2017) [6]. We evaluate using the WEAT test before and after linear projection debiasing in Table 6. For each of these tests, we use half of the names in each category for finding the bias direction and the other half for WEAT testing. This selection is done arbitrarily and the scores are averaged over 3 such selections.

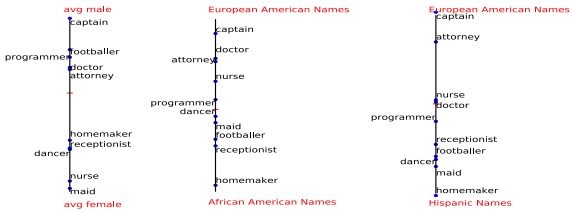

Figure 4: Gender and racial bias in the embedding

Table 6: WEAT positive-negative test scores before and after debiasing

	Before Debiasing	After Debiasing
EA-AA	1.803	0.425
EA-H	1.461	0.480
Youth-Aged	0.915	0.704

More qualitatively, as a result of the dampening of bias, we see that biased words like other names belonging to these specific demographic groups, slang words, colloquial terms like `latinos` are removed from the closest 10% words. This is beneficial since the distinguishability of demographic characteristics based on names is what shows up in these different ways like in occupational or financial bias.

Age-associated names. We observed that names can be masked carriers of age too. Using the database for names through time [1] and extracting the most common names from early 1900s as compared to late 1900s and early 2000s, we find a correlation between these names and age related words. In the full version [7] we have demonstrated how there is a clear correlation between age and names. Bias in this case does not show up in professions as clearly as in gender but in terms of association with positive and negative words [6]. We again evaluate using a WEAT test in Table 6, the bias before and after debiasing the embedding.

7 DISCUSSION

Different types of bias exist in textual data. We see here how a simple linear projection of word embeddings away from the bias direction effectively corrects that bias without affecting other intrinsic properties. Further, using names we can detect the direction of different kinds of bias in word embeddings, and thus, remove it with linear projection.

References

[1] https://www.ssa.gov/oact/babynames/.

[2] T Bolukbasi, K W Chang, J Zou, V Saligrama, and A Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *ACM Transactions of Information Systems*, 2016.

[3] T Bolukbasi, K W Chang, J Zou, V Saligrama, and A Kalai. Quantifying and reducing bias in word embeddings. 2016.

[4] Tim Brennan, William Dieterich, and Beate Ehret. Evaluating the predictive validity of the compas risk and needs assessment system. *Criminal Justice and Behavior*, 36(1):21–40, 2009.

[5] Kaylee Burns, Lisa Anne Hendricks, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. *CoRR*, abs/1803.09797, 2018.

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

[7] Sunipa Dev and Jeff Phillips. Attenuating Bias in Word Vectors. *arXiv e-prints*, 2019.

[8] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[9] L Finkelstein, E Gabrilovich, Y Matias, E Rivlin, Z Solan, G Wolfman, and etal. Placing search in context : The concept revisited. In *ACM Transactions of Information Systems*, volume 20, pages 116–131, 2002.

[10] F Hill, R Reichart, and A Korhonen. Simlex-999 : Evaluating semantic models with (genuine) similarity estimation. In *Computational Linguistics*, volume 41, pages 665–695, 2015.

[11] Amir E. Khandani, Adlar J. Kim, and Andrew Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010.

[12] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS*, 2013.

[13] Omer Levy and Yoav Goldberg. Linguistic regularities of sparse and explicit word representations. In *CoNLL*, 2014.

[14] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. Technical report, arXiv:1301.3781, 2013.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. 2014.

[18] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *CoRR*, abs/1707.09457, 2017.