Analyzing Correlations Between Intrinsic and Extrinsic Bias Metrics of Static Word Embeddings With Their Measuring Biases Aligned

Anonymous ACL submission

Abstract

We examine the abilities of intrinsic bias metrics of static word embeddings to predict whether Natural Language Processing (NLP) systems exhibit biased behavior. A word embedding is one of the fundamental NLP technologies that represents the meanings of words through real vectors, and problematically, it also learns social biases such as stereotypes. An intrinsic bias metric measures bias by examining a characteristic of vectors, while an extrinsic bias metric checks whether an NLP system trained with a word embedding is biased. A previous study found that a common intrinsic bias metric usually does not correlate with extrinsic bias metrics. However, the intrinsic and extrinsic bias metrics did not measure the same bias in most cases, which makes us question whether the lack of correlation is genuine. In this paper, we extract characteristic words from datasets of extrinsic bias metrics and analyze correlations with intrinsic bias metrics with those words to ensure both metrics measure the same bias. We observed moderate to high correlations with some extrinsic bias metrics but little to no correlations with the others. This result suggests that intrinsic bias metrics can predict biased behavior in particular settings but not in others.

1 Introduction

We re-analyze correlations between two main schemes to measure bias in static word embeddings, i.e., "intrinsic" and "extrinsic" bias metrics (Goldfarb-Tarrant et al., 2021). A word embedding encodes stereotypical association, or bias, from a training corpus (Bolukbasi et al., 2016; Caliskan et al., 2017). An intrinsic bias metric takes only word embeddings into account to measure bias. On the other hand, an extrinsic bias metric trains NLP systems with word embeddings and measures the bias of word embeddings by checking if the systems behave differently to different people. A biased NLP system works inadequately for one group of people compared to another. To make sure that an NLP system works for everyone, we need to improve extrinsic bias metrics.

Goldfarb-Tarrant et al. (2021) found that research on reducing word embedding bias heavily relies on the assumption that lowering intrinsic bias metrics improves extrinsic bias metrics but that no research confirmed this assumption. In response, they analyzed the relation between a common intrinsic bias metric WEAT (Caliskan et al., 2017) and extrinsic bias metrics by generating and measuring the bias of word embeddings with various bias levels, from debiased ones to over-biased ones. They found that the intrinsic and extrinsic bias metrics usually do not correlate. They recommend researchers and practitioners not rely on intrinsic bias metrics and instead measure bias with extrinsic metrics and focus on mitigating harms caused by bias.

However, in their experiment, WEAT and extrinsic bias metrics do not measure the same bias in most cases. For example, WEAT 7 (Caliskan et al., 2017), one of the word sets they used to measure WEAT, measures bias such as "Boys are good at math, while girls are good at literature." This bias is not matched with the one of the WinoBias (Zhao et al., 2018) extrinsic bias metric, which is like "Doctors are men, while secretaries are women." Although both are about gender bias, the former is about gender bias in school subjects, while the latter is about gender bias in professions. This mismatch makes us question whether the lack of correlation is genuine.

Cao et al. (2022) analyzed correlations between intrinsic and extrinsic bias metrics of language models with their measuring biases matched. They extracted characteristic words from a dataset for an extrinsic bias metric to make intrinsic and extrinsic bias metrics measure the same bias. An intrinsic bias metric was measured with those words. Cao

et al. (2022) measured the bias of 19 pre-trained language models with bias metrics. However, none of the language models underwent a debiasing procedure. It is necessary to generate many data points with various bias levels to analyze correlations between metrics.

In this study, we make intrinsic and extrinsic bias metrics measure the same bias and re-analyze their correlations, using word embeddings with various bias levels as data points. Word embeddings are smaller and easier to train than language models, which are suitable properties to generate many data points with various bias levels. We extract bias-representing word sets from a dataset for an extrinsic bias metric and analyze correlations with intrinsic bias metrics measured with those word sets. For example, when analyzing correlations between intrinsic bias metrics and the WinoBias extrinsic bias metric, we extract a stereotypically male job word set, stereotypically female job word set, male word set, and female word set and measure the intrinsic bias metrics with them.

We observed moderate to high correlations with some extrinsic bias metrics but little to no correlations with the others. This result suggests that intrinsic bias metrics can be trusted in particular settings but not in others.

2 Background

In this section, we describe bias metrics of which we analyze correlations and related work.

2.1 WEAT

WEAT is a common intrinsic bias metric to measure stereotypical association (e.g., "physician" and "man" or "secretary" and "woman") (Caliskan et al., 2017). For example, suppose we want to measure gender bias in occupations. WEAT requires word sets T_1 and T_2 ($|T_1| = |T_2|$) that represent two target concepts of bias (stereotypically male and female jobs). They are called target words. It also requires *attribute words* A_1 and A_2 (male and female words) that are stereotypically associated with each target word set even though they should not. "physician" should be as close to "woman" as to "man." Given those words, WEAT is defined as $\sum_{t \in T_1} s(t, A_1, A_2) - \sum_{t \in T_2} s(t, A_1, A_2)$ where $s(t, A_1, A_2) = \frac{1}{|A_1|} \sum_{a \in A_1} \cos(t, a) - \frac{1}{|A_2|} \sum_{a \in A_2} \cos(t, a)$. By normalizing the WEAT value, one can obtain the effect size, i.e., a value indicating how different target word sets are in terms of associativity with attribute words:

$$\frac{\frac{1}{|T_1|} \sum_{t \in T_1} s(t, T_1, T_2) - \frac{1}{|T_2|} \sum_{t \in T_2} s(t, A_1, A_2)}{\text{std_dev}_{t \in T_1 \cup T_2} s(t, A_1, A_2)}.$$

Caliskan et al. (2017) also defined word sets, WEAT 1 to 10, with which WEAT is measured.

2.2 RNSB

The Relative Negative Sentiment Bias (RNSB) (Sweeney and Najafian, 2019) is an intrinsic bias metric to assess how much negative sentiment is associated with identity terms such as demonyms. Let A_1 and A_2 be positive and negative sentiment word sets, respectively, and $\{t_1, ..., t_N\}$ be identity terms representing N types of people. It first trains a logistic regression sentiment classifier f with A_1 and A_2. After training, it predicts negative sentiment probability $\overline{f}(t_i)$ for each identity term t_i. By normalizing $\overline{f}(t_1), ..., \overline{f}(t_N)$, we can obtain a probabilistic distribution

$$D = \left(\frac{\overline{f}(t_1)}{\sum_i \overline{f}(t_i)}, ..., \frac{\overline{f}(t_N)}{\sum_i \overline{f}(t_i)} \right) \quad (1)$$

over identity terms. No identity term should be predicted as more negative than any other term, and the distribution D should be uniform. The RNSB is the KL divergence of D from a uniform distribution U over N identity terms.

Badilla et al. (2020) extended the RNSB to incorporate target and attribute words. Given attribute word sets A_1 and A_2[1] and target word sets $T_1, ..., T_N$, the RNSB trains logistic regression f with A_1 as positive examples and A_2 as negative ones. Then, the RNSB is the KL divergence of a probabilistic distribution over $w \in T_1 \cup ... \cup T_N$ from a uniform distribution. However, in this extension, the distinction of the target word sets has no effect. Even if we merge all target words and create one big target word set T, the RNSB is unchanged.

Instead of the probabilistic distribution over all target words, we consider a distribution over target word sets. Let $\overline{f}(T_i)$ be the average probability that word w in T_i is a negative example. We can measure the RNSB by replacing $\overline{f}(t_i)$ with $\overline{f}(T_i)$ in Equation 1. However, we can simplify the calculation in our experiment. We only have cases where $N = 2$, and thus how different D is from

[1] Note that A_1 and A_2 are no longer restricted to positive and negative sentiment words.

the uniform distribution is represented by

$$\frac{\overline{f}(T_2)}{\overline{f}(T_1)+\overline{f}(T_2)} - \frac{\overline{f}(T_1)}{\overline{f}(T_1)+\overline{f}(T_2)} = \frac{\overline{f}(T_2)-\overline{f}(T_1)}{\overline{f}(T_1)+\overline{f}(T_2)}.$$

This simplifies the calculation and gives us a more fine-grained metric. Imagine a word embedding v that is overly debiased. That is, A_1 is strongly associated with T_2, and A_2 is with T_1. This makes $\overline{f}(T_1)$ higher than $\overline{f}(T_2)$. The KL-divergence-based RNSB only tells us v is heavily biased, while our RNSB tells us it is anti-stereotypically biased.

2.3 WinoBias

WinoBias (Zhao et al., 2018) is a dataset for bias evaluation of coreference resolution systems. It contains sentences representing pro- and anti-stereotypical situations and measures the bias of the systems by comparing outputs for both situations. Goldfarb-Tarrant et al. (2021) trained coreference resolution systems with word embeddings and let the bias scores of the systems be bias scores of the word embeddings. In sentences in pro-stereotypical sets (e.g., "The physician hired the secretary because he was overwhelmed with clients."), antecedents of third-person pronouns are occupations that are stereotypically associated with the genders of the pronouns. In anti-stereotypical sets, the genders of the pronouns are swapped. If a coreference resolution system does not have gender bias, its outputs for both sentences should be the same. Let p_{pro} and p_{anti} be performances (precision or recall) of the system on pro- and anti-stereotypical sets, respectively. Goldfarb-Tarrant et al. (2021) defined a bias score as $p_{\text{pro}} - p_{\text{anti}}$. WinoBias has two types, and sentences in Type 2 include syntactic cues to resolve coreferences (e.g., "The secretary called the physician and told him about a new patient.").

2.4 Hate Speech Detection

Goldfarb-Tarrant et al. (2021) used hate speech detection (HSD) for extrinsic bias metric. They extracted and annotated 15,000 tweets as targeting male, female, or neutral from the abusive tweet dataset of Founta et al. (2018). Letting p_{male} and p_{female} be performances of a hate speech detector on tweets targeting male and female, respectively, they defined a bias score of a word embedding as $p_{\text{male}} - p_{\text{female}}$. In addition to that, we measure the difference $p_{\text{w}} - p_{\text{aa}}$ between performances for white-sounding tweets p_{w} and African-American-sounding tweets p_{aa}. African-American English is often misunderstood as hateful (Sap et al., 2019), and therefore, comparing the HSD performances is a good extrinsic bias metric for racial bias.

2.5 Related Work

Goldfarb-Tarrant et al. (2021) found that research on reducing word embedding bias heavily relies on the assumption that reducing intrinsic bias improves extrinsic bias metrics but that no research confirmed this. To test this assumption, they analyzed the relation between the WEAT effect size and extrinsic bias metrics under various conditions (two word embedding algorithms, two downstream tasks, and two languages, English and Spanish). They generated word embeddings with various bias levels, measured the biases with WEAT and extrinsic bias metrics, and analyzed their relations by calculating Pearson's r and making scatter plots. As a result of the experiment, Goldfarb-Tarrant et al. (2021) found that WEAT does not correlate with extrinsic bias metrics in most cases. However, the biases that WEAT measures and the ones that extrinsic bias metrics measure were not aligned well, which makes us question whether the lack of correlation is genuine. In our study, we close this gap and re-analyze correlations.

For language models, Cao et al. (2022) analyzed correlations between intrinsic and extrinsic bias metrics with adjustments of several misalignments. For example, when measuring correlations with toxicity-related extrinsic bias metrics, they manually chose toxic and anti-toxic words from word clouds generated from extrinsic bias metric dataset Jigsaw Toxicity (cjadams et al., 2019), and measured CEAT (Guo and Caliskan, 2021) with them. However, their analysis is based on 19 pre-trained language models, none of which underwent a debiasing procedure. It is necessary to generate many data points with various degrees of bias to analyze correlations between metrics. In our study, we generate many word embeddings with various bias levels, following Goldfarb-Tarrant et al. (2021), and analyze correlations. Even though it is less potent than a large language model, a word embedding is smaller and easier to train. It is suitable for generating many data points with various bias levels.

Orgad et al. (2022) also examined the relationship between intrinsic and extrinsic bias metrics of

language models. They proposed a new intrinsic bias metric, "compression," and showed it correlates well with extrinsic bias metrics to some degree. However, they considered only gender bias. We take racial bias into account as well.

3 Method

Given a dataset to measure an extrinsic bias metric, we extract target and attribute word sets from it that represent bias the extrinsic bias metric measures. By feeding those word sets to an intrinsic bias metric, we can make it measure the same bias as the extrinsic bias metric. Then, we analyze the correlation between the metrics, following Goldfarb-Tarrant et al. (2021). Ensuring that both metrics measure the same bias is essential to analyze the correlation accurately. Suppose it does not hold, and the correlation is weak; we cannot conclude whether it is due to a misalignment of biases measured by both metrics or a fundamental difference between them.

From WinoBias, we extract stereotypical male/female occupations as target words and $\{$"he", "him"$\}$ and $\{$"she", "her"$\}$ as attribute words.[2]

An HSD extrinsic bias metric measures bias by comparing performances for two groups (in our experiment, male-targeting tweets vs. female-targeting tweets (Goldfarb-Tarrant et al., 2021) and white-sounding tweets vs. African-American-sounding tweets). Its measuring bias is that one group is associated with hatefulness. We extract words representing the two groups as target words and hateful and unhateful words as attribute words. We extract those words based on PMIs (Fano, 1963). Given an HSD dataset consisting of tweets with two labels (one indicating hatefulness and the other showing a group of people), we first calculate PMIs between an event where a tweet is labeled as $l \in \{\text{HS}, \text{NON-HS}\}$ (HS stands for "hate speech.") and an event where the word w appears in a tweet. Let it be $\text{PMI}(l, w)$. To calculate $\text{PMI}(l, w)$, we assume the Naïve Bayes generative model, where a tweet is labeled as l with a probability of μ_l and its tokens are sampled from a categorical distribution parameterized with $\boldsymbol{\theta}_l$. After estimating the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$, we calculate $\text{PMI}(l, w)$ as

$$\log_2 \frac{\theta_{l,w}}{\sum_{l'} \mu_{l'} \theta_{l',w}}.$$

We extract candidate hateful attribute words $w_1, ..., w_{40}$ where $\text{PMI}(\text{HS}, w_i)$ is the i-th largest. We do not want to pick up rare words. Therefore, we filter out words appearing in less than ten tweets in the dataset, and when candidate attribute words have the same PMI, the more frequent word is chosen. We then manually select hateful attribute words from the candidates. Non-hateful attribute words and target words are extracted in similar ways. (When extracting, for example, men and women target words, let $l \in \{\text{MALE}, \text{FEMALE}\}$.) A detailed selection process is explained in Appendix A.

4 Experiment

Following Goldfarb-Tarrant et al. (2021), we generate word embeddings with various bias levels, measure their bias with intrinsic and extrinsic bias metrics, and analyze their correlations.

4.1 Bias Metrics

We employ the RNSB (intrinsic bias metric) and the racial performance gap of HSD (extrinsic bias metric), in addition to metrics used in the English experiment of Goldfarb-Tarrant et al. (2021) (the WEAT effect size, WinoBias, and the gender performance gap of HSD). The RNSB is similar to extrinsic bias metrics in that both train a machine learning model. We include the RNSB because it might correlate well with extrinsic bias metrics. The RNSB is also similar to the intrinsic bias metric "compression" (Orgad et al., 2022) in that both measure bias with classifiers. Details about measuring the RNSB are in Appendix K.

To measure the racial performance gap, we need to generate white-sounding and African-American-sounding tweet sets. We use the mixed-membership demographic-language model developed by Blodgett et al. (2016). It receives a list of word tokens and predicts the proportions of dialects of each race and ethnicity (white, African-American, Hispanic, and Asian). We take the argmax of these proportions for each tweet in the test dataset to determine race or ethnicity.

For extrinsic bias metrics, Goldfarb-Tarrant et al. (2021) compared precisions and recalls. We also compare the F1 scores to see the overall performance difference.

[2] Stereotypical male occupations contain a two-word expression (i.e., "construction worker"). Its word embedding is an average of "construction" and "worker."

NLP models trained for the extrinsic bias metrics are the same as in Goldfarb-Tarrant et al. (2021). The coreference resolution model is proposed by Lee et al. (2017), a high-performance, end-to-end neural model. We use Goldfarb-Tarrant et al. (2021)'s implementation. It is trained with OntoNotes (Weischedel et al., 2017) 5.0, split into training/validation/test datasets as in CoNLL-2012 Shared Task (Pradhan et al., 2012). The HSD model is based on the "CNN-static" of Kim (2014), a simple text classifier applying several convolutional layers in parallel. Hyperparameter values are in Appendix L. It is trained with the abusive tweet dataset of Founta et al. (2018). Dataset preprocessing is explained in Appendix M. Basic statistics of the datasets are shown in Appendix H.

We measure intrinsic bias metrics with the extracted words and WEAT words to see the influence of word sets. When measuring gender bias, we use WEAT 6, 7, and 8, which were used in Goldfarb-Tarrant et al. (2021). We use WEAT 3, 4, and 5 to measure racial bias. Goldfarb-Tarrant et al. (2021) substituted general male and female words for male and female names in WEAT 6 because these names do not appear in WinoBias. We adopt the substitution. Since our training corpora of word embeddings are all lowercased, we lowercase WEAT words when measuring intrinsic bias metrics. Some target words in WEAT word sets do not appear in a training corpus of word embeddings. We remove those words and an equal number of words from the other target word set so that both target word sets have the same number of words. Appendix B shows WEAT words used in our experiment.

Table 1 summarizes the word sets used in our experiment.

4.2 Training of Word Embeddings

Training settings for word embeddings are similar to Goldfarb-Tarrant et al. (2021). We train Skip-gram word2vec with negative sampling (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017) word embeddings. Word embeddings for coreference resolution are trained on Wikipedia and on Twitter data for HSD. A detailed explanation is given in Appendix F.

4.3 Bias Modification

To modify bias, we need word sets to define bias as in intrinsic bias metrics. Goldfarb-Tarrant et al. (2021) obtained these word sets by merging word sets with which intrinsic bias metrics were measured, expanding them following Lauscher et al. (2020), and removing inappropriate words manually. Given a word embedding v (not an embedding of which we measure bias), merged target word sets T_1 and T_2, and merged attribute word sets A_1 and A_2, for each word w in a word set, its K closest words $\{w_i\}_i$ in v in cosine similarity terms are added to the word set. If w_i is already in $T_1 \cup T_2 \cup A_1 \cup A_2$, it is skipped. Details about the hyperparameters we chose and removing inappropriate words are in Appendix I. Table 2 summarizes with which word sets word embeddings are modified and evaluated.

Following Goldfarb-Tarrant et al. (2021), we ran the dataset balancing (Dixon et al., 2018) and ATTRACT-REPEL (Mrkšić et al., 2017) to generate word embeddings with various bias levels. They ran the algorithms several times until certain conditions were met. We create word embeddings with various degrees of bias by changing the parameters of the algorithms. This method allows us to create more word embeddings. In total, 45 word embeddings are generated. This is more than twice as many data points as in Goldfarb-Tarrant et al. (2021) and Cao et al. (2022) (which analyzed correlations for language models).

The dataset balancing makes a dataset more balanced (e.g., in terms of frequencies of male and female entities) so that a model trained on it is less biased. Goldfarb-Tarrant et al. (2021) reduced stereotypical sentences (e.g., "He is a physician.") to reduce bias and anti-stereotypical sentences to enhance bias. Here, a stereotypical sentence contains words in T_1 and A_1 or T_2 and A_2. (T_1 and T_2 are target word sets, and A_1 and A_2 are attribute word sets.) An anti-stereotypical sentence contains words in T_1 and A_2 or T_2 and A_1. When running dataset balancing on the Twitter corpus, we regard a tweet as a sentence. We introduce a parameter $p \in [0, 1)$ (a sampling probability) to the algorithm. For each (anti-)stereotypical sentence, we keep it with probability p and discard it with probability $1 - p$. We run the algorithm with $p \in \{0.0, 0.1, 0.2, ..., 0.9\}$ for both debiasing and over-biasing.

ATTRACT-REPEL is an algorithm designed to improve the semantics of word embeddings. It makes synonym pairs closer and antonym pairs far away. Goldfarb-Tarrant et al. (2021) re-purposed it for bias modification. Given target words T_1 and T_2 and attribute words A_1 and A_2, they debiased

Name	T_1	T_2	A_1	A_2
WinoBias word sets	Stereotypical male jobs	Stereotypical female jobs	Male	Female
HSD gender bias word sets	Male	Female	Hateful	Unhateful
HSD racial bias word sets	White-sounding words	African-American-sounding words	Unhateful	Hateful
WEAT 3, 4, and 5	European American names	African American names	Pleasant	Unpleasant
WEAT 6	Career	Family	Male	Female
WEAT 7	Math	Arts	Male	Female
WEAT 8	Science	Arts	Male	Female

Table 1: Summarization of word sets we used. T_1 and T_2 are target words, and A_1 and A_2 are attribute words. The original WEAT 6 has male and female target words and career and family attribute words. We swap them, as Goldfarb-Tarrant et al. (2021) did so that male and female words are attribute words as in WEAT 7 and 8.

Modification word sets	Evaluation word sets
Expansion of merged WEAT 6, 7, and 8	WEAT 6 WEAT 7 WEAT 8
Expansion of WinoBias word sets	WinoBias word sets
Expansion of HSD gender bias word sets	HSD gender bias word sets
Expansion of merged WEAT 3, 4, and 5	WEAT 3 WEAT 4 WEAT 5
Expansion of HSD racial bias word sets	HSD racial bias word sets

Table 2: Summarization of word sets with which word embeddings are modified and evaluated.

word embeddings by setting $T_1 \times A_2 \cup T_2 \times A_1$ as synonym pairs and $T_1 \times A_1 \cup T_2 \times A_2$ as antonym pairs. When over-biasing, synonym and antonym pairs were swapped. We run the algorithm while changing the parameters. The combinations of the parameters are in Appendix O.

5 Results and Discussion

Table 3, 4, 5, and 6 show Spearman's correlations between intrinsic and extrinsic bias metrics. Figure 1 shows a couple of scatter plots.[3]

When WEAT is measured with the WinoBias word sets, it shows strong correlations with WinoBias contrary to the previous study (Goldfarb-Tarrant et al., 2021). It means WEAT can roughly predict the biased behavior of coreference resolution systems. However, when we use the WEAT word sets, which were used in Goldfarb-Tarrant et al. (2021), the correlation disappears. This result shows the importance of matching measuring biases of intrinsic and extrinsic bias metrics. A similar result for language models was reported in Cao et al. (2022).

The RNSB shows moderate correlations with WinoBias. However, these correlations are unreliable for estimating the WinoBias metric from the RNSB because the WinoBias metric varies greatly even for a similar RNSB value (the top-right of Figure 1).

Correlations with the HSD extrinsic bias metrics are small. Unsurprisingly, the F1 difference has small correlations with intrinsic bias metrics. For example, when racial intrinsic bias metrics increase, that means associations between white-sounding words and unhateful ones and between African-American-sounding words and hateful ones increase. This makes the HSD system predict white-sounding tweets as unhateful and African-American-sounding ones as hateful. That results in increase of precision difference $p_{\text{w}} - p_{\text{aa}}$, decrease of recall difference $r_{\text{w}} - r_{\text{aa}}$, and relatively stable F1 difference $f_{\text{w}} - f_{\text{aa}}$.

Following this logic, positive correlations of precision difference and negative correlations of recall

[3]Refer to the supplementary material for all the scatter plots.

		Word sets with which intrinsic bias metrics are measured							
		Extracted		WEAT 6		WEAT 7		WEAT 8	
Extrinsic bias metric		WEAT	RNSB	WEAT	RNSB	WEAT	RNSB	WEAT	RNSB
WB type 1	Precision diff	0.84	0.56	0.09	0.08	0.11	0.11	0.07	0.07
	Recall diff	0.84	0.58	0.04	0.05	0.07	0.08	0.03	0.04
	F1 diff	0.83	0.56	0.07	0.06	0.1	0.1	0.06	0.06
WB type 2	Precision diff	0.91	0.6	−0.11	−0.1	−0.13	−0.1	−0.16	−0.11
	Recall diff	0.9	0.59	−0.09	−0.07	−0.1	−0.05	−0.14	−0.05
	F1 diff	0.91	0.6	−0.09	−0.07	−0.11	−0.06	−0.14	−0.07
HSD	Precision diff	−0.11	−0.17	−0.03	0.04	−0.08	0.04	−0.07	0.01
	Recall diff	0.33	0.26	−0.15	−0.13	−0.08	−0.13	−0.14	−0.17
	F1 diff	0.21	0.14	−0.11	−0.08	−0.08	−0.08	−0.12	−0.13

Table 3: Gender bias results with word2vec. WB stands for WinoBias.

		Word sets with which intrinsic bias metrics are measured							
		Extracted		WEAT 6		WEAT 7		WEAT 8	
Extrinsic bias metric		WEAT	RNSB	WEAT	RNSB	WEAT	RNSB	WEAT	RNSB
WB type 1	Precision diff	0.91	0.66	−0.1	−0.04	−0.08	−0.08	−0.1	−0.1
	Recall diff	0.91	0.67	−0.09	0	−0.1	−0.04	−0.11	−0.06
	F1 diff	0.91	0.66	−0.1	−0.03	−0.1	−0.06	−0.12	−0.08
WB type 2	Precision diff	0.88	0.57	−0.17	−0.13	−0.21	−0.21	−0.2	−0.16
	Recall diff	0.89	0.58	−0.2	−0.09	−0.22	−0.17	−0.21	−0.14
	F1 diff	0.89	0.58	−0.17	−0.09	−0.2	−0.17	−0.19	−0.13
HSD	Precision diff	−0.14	−0.08	−0.23	−0.09	−0.19	−0.07	−0.25	−0.1
	Recall diff	0.21	0.19	0.22	0.06	0.24	0.04	0.23	0.01
	F1 diff	0.01	0.05	0	−0.07	0.03	−0.05	0.01	−0.07

Table 4: Gender bias results with fastText

		Word sets with which intrinsic bias metrics are measured							
		Extracted		WEAT 3		WEAT 4		WEAT 5	
Extrinsic bias metric		WEAT	RNSB	WEAT	RNSB	WEAT	RNSB	WEAT	RNSB
HSD	Precision diff	0.24	0.29	−0.03	0.13	−0.06	0.16	−0.07	0.24
	Recall diff	−0.2	−0.15	0.02	−0.06	0.02	−0.07	0.06	−0.01
	F1 diff	−0.04	0.12	−0.04	−0.05	−0.05	−0.05	−0.03	0.04

Table 5: Racial bias results with word2vec

		Word sets with which intrinsic bias metrics are measured							
		Extracted		WEAT 3		WEAT 4		WEAT 5	
Extrinsic bias metric		WEAT	RNSB	WEAT	RNSB	WEAT	RNSB	WEAT	RNSB
HSD	Precision diff	0.09	0.12	−0.17	−0.01	−0.17	0	−0.12	0.14
	Recall diff	−0.13	0.05	0.08	−0.02	0.08	0	0.03	−0.01
	F1 diff	−0.23	0.12	−0.01	−0.03	−0.02	0.03	−0.03	0.12

Table 6: Racial bias results with fastText

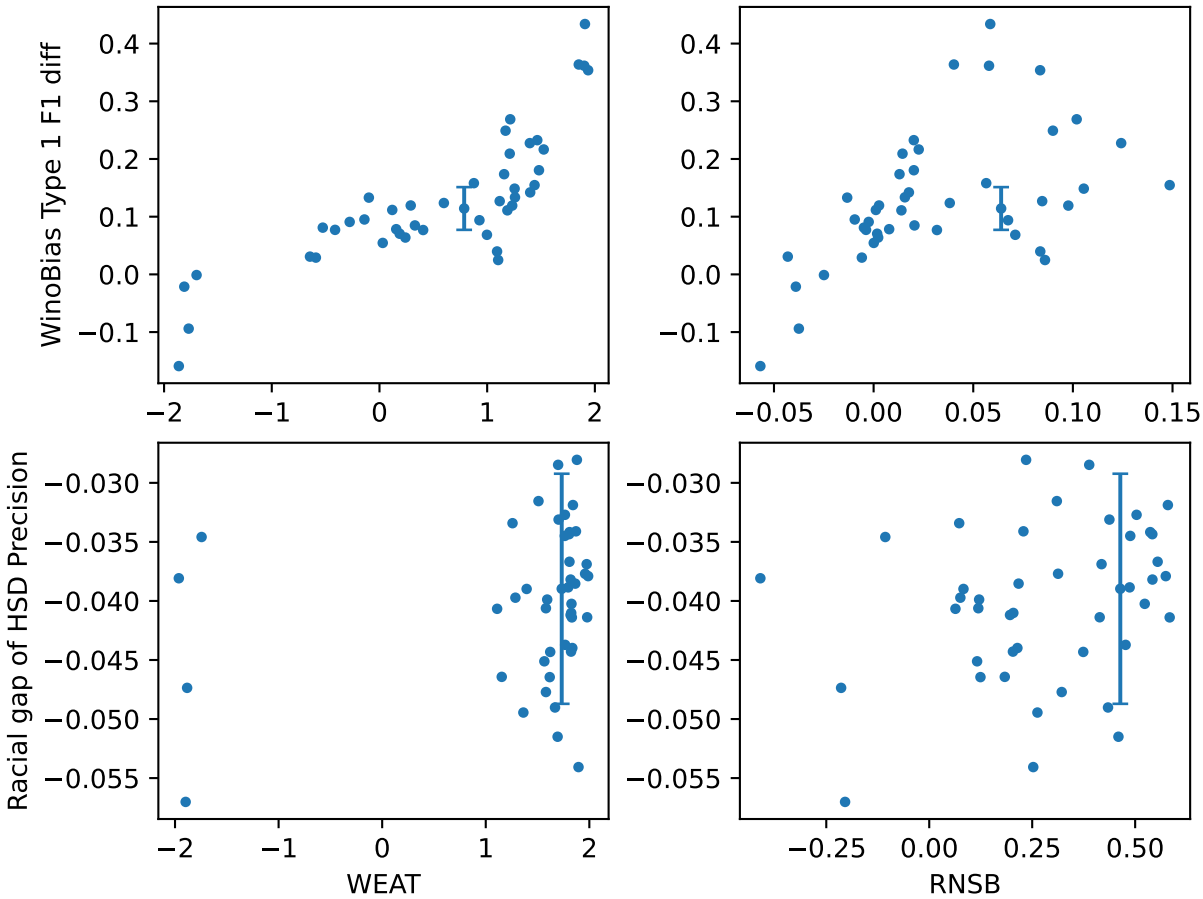

Figure 1: Correlations between intrinsic bias metrics (x-axis) and extrinsic bias metrics (y-axis). The intrinsic bias metrics are measured with the corresponding extracted word sets. The word embedding algorithm used is word2vec. For original (not bias-modified) word embeddings, we train NLP models ten times so that we can calculate standard deviations of extrinsic bias metrics, which are shown as error bars.

difference are expected for racial bias. For gender bias, the opposite trend is expected since hateful words are associated with male terms, and unhateful words are with female terms. (Compare the HSD gender bias word sets and racial bias word sets in Table 1.)

The results do not match our expectations well. The correlations are small. The influence of word sets is also unexpected. We observed stronger correlations for gender and racial bias in the word2vec setting when intrinsic bias metrics were measured with the extracted word sets. However, WEAT for gender bias in the fastText setting shows the opposite trend.

One possible hypothesis for the lack of correlation is that there is a fundamental difference between intrinsic bias metrics and the HSD bias metrics. In the HSD, an occurrence of a hateful word does not necessarily mean a tweet is hate speech. Intrinsic bias metrics, on the other hand, focus on only word embeddings of given words (e.g., hateful words). This difference might prevent us from matching their measuring biases. This hypothesis also explains why the WinoBias metric strongly correlates with intrinsic bias metrics. The WinoBias metric focuses on target and attribute words, similar to intrinsic bias metrics, and observes which target word is linked to an attribute word.

6 Conclusions

In this study, we re-analyzed Goldfarb-Tarrant et al. (2021) by making intrinsic and extrinsic bias metrics measure the same bias. We extracted target and attribute words from extrinsic bias metrics and measured intrinsic bias metrics with them. We observed moderate to high correlations with the WinoBias extrinsic bias metrics in the experiment but little to no correlations with the HSD metrics. This result suggests that intrinsic bias metrics can predict a biased behavior in coreference resolution but not in HSD.

7 Ethical Considerations

Although we showed strong correlations between WEAT and the WinoBias extrinsic bias metrics, researchers and practitioners should not replace the WinoBias extrinsic bias metrics with WEAT. The purpose of our study is to assess the reliability of intrinsic bias metrics. Even when a word embedding has the WEAT metric of 0, the WinoBias extrinsic bias metric shows an F1 difference of around 0.1. (the top-left of Figure 1) Replacing the WinoBias extrinsic bias metrics with WEAT might make us underestimate the bias.

8 Limitations

We could not show why the intrinsic bias metrics correlate with the WinoBias extrinsic bias metrics but not the HSD bias metrics. Analyzing correlations with more extrinsic bias metrics might be necessary to discover why.

References

Giuseppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor.

Pablo Badilla, Felipe Bravo-Marquez, and Jorge Pérez. 2020. Wefe: The word embeddings fairness evaluation framework. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 430–436. International Joint Conferences on Artificial Intelligence Organization.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marianne Bertrand and Sendhil Mullainathan. 2004. Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination. *American Economic Review*, 94(4):991–1013.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc., Sebastopol, CA.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

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

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 561–570, Dublin, Ireland. Association for Computational Linguistics.

cjadams, Daniel Borkan, inversion, Jeffrey Sorensen, Lucas Dixon, Lucy Vasserman, and nithum. 2019. Jigsaw unintended bias in toxicity classification.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 67–73, New York, NY, USA. Association for Computing Machinery.

Robert M. Fano. 1963. *Transmission of information: a statistical theory of communications*. M.I.T. Press, Cambridge, MA.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(11).

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform.

Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya, and Adam Lopez. 2021. Intrinsic bias metrics do not correlate with application bias. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1926–1940, Online. Association for Computational Linguistics.

Anthony G. Greenwald, Debbie E. McGhee, and Jordan L. K. Schwartz. 1998. Measuring individual differences in implicit cognition: The implicit association test. *Journal of Personality and Social Psychology*, 74(6):1464–1480.

Wei Guo and Aylin Caliskan. 2021. Detecting emergent intersectional biases: Contextualized word embeddings contain a distribution of human-like biases. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 122–133, New York, NY, USA. Association for Computing Machinery.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python. Original-date: 2014-07-03T15:15:40Z.

John D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Internet Archive. Internet archive: Digital library of free & borrowable books, movies, music & wayback machine.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Anne Lauscher, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2020. A general framework for implicit and explicit debiasing of distributional word vector spaces. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(0505):8131–8138.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Microsoft Corporation. 2021. Bling fire.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Brian A. Nosek, Mahzarin R. Banaji, and Anthony G. Greenwald. 2002a. Harvesting implicit group attitudes and beliefs from a demonstration web site. *Group Dynamics: Theory, Research, and Practice*, 6(1):101–115.

Brian A. Nosek, Mahzarin R. Banaji, and Anthony G. Greenwald. 2002b. Math = male, me = female, therefore math $\neq$ me. *Journal of Personality and Social Psychology*, 83(1):44–59.

Hadas Orgad, Seraphina Goldfarb-Tarrant, and Yonatan Belinkov. 2022. How gender debiasing affects internal model representations, and why it matters. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2602–2628, Seattle, United States. Association for Computational Linguistics.

The pandas development team. 2023. pandas-dev/pandas: Pandas.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett. 2019. Pytorch: An imperative style, high-performance deep learning library.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Python. 1990. Cpython. `https://github.com/python/cpython`.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop*

on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Evan T. R. Rosenman, Santiago Olivella, and Kosuke Imai. 2023. Race and ethnicity data for first, middle, and surnames. *Scientific Data*, 10(1):299. Number: 1 Publisher: Nature Publishing Group.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Jason Scott. 2017. Archiveteam json download of twitter stream 2017-04 (Internet Archive).

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*, pages 1662–1667, Florence, Italy. Association for Computational Linguistics.

The Wikimedia Foundation. 2001. Wikipedia.

TorchText maintainers and contributors. 2016. Torchtext. https://github.com/pytorch/text.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Ralph M. Weischedel, Eduard H. Hovy, Mitchell P. Marcus, and Martha Palmer. 2017. Ontonotes : A large training corpus for enhanced processing.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Note: The following appendixes contain offensive language.

A Word Sets Extraction

We select target and attribute words from the candidates. The average number of target and attribute words in the corresponding WEAT word sets decides how many target and attribute words are chosen. When extracting words for gender bias, the corresponding WEAT word sets are WEAT 6, 7, and 8, designed to measure gender bias. For the same reason, the corresponding WEAT word sets are WEAT 3, 4, and 5 when extracting words for racial bias.

We extracted the following candidate words from the gender HSD dataset used in Goldfarb-Tarrant et al. (2021) (Table 7). We removed "flourish," "yall," and "females" from hateful words for not representing hate. We removed "slut," "bitches," and "bitch" for associations with females. We then chose the first eight words in candidates for unhateful and hateful attribute words. We chose "he's," "boy," "he," "him," and "king" as male target words and "women's," "lady," and "wife," as female target words.[4] For each word, we added the word of the opposite gender. Table 8 shows the final target and attribute words.

We extracted the following candidate words from the racial HSD dataset (Table 9). To choose white and African-American target words, we ran the mixed-membership demographic-language model developed by Blodgett et al. (2016). It receives a list of word tokens and predicts the proportions of dialects of each race and ethnicity (white, African-American, Hispanic, and Asian). We took the argmax of these proportions for each candidate word to determine its race or ethnicity. We chose words in Table 10 as the target and attribute words.

B WEAT Words Used in Our Experiment

B.1 WEAT 3

The original word sets are taken from Greenwald et al. (1998). Some of the African American names do not appear in our processed Twitter corpus. We removed them. To make target words equal in size, we removed the same number of names from the European American names randomly.

- European American names (target words): Adam Roger Alan Frank Justin Ryan Andrew

[4] We did not choose "guy," "babe," and "bitches" because their opposite sex counterparts are unclear.

Male target words	Female target words	Hateful attribute words	Unhateful attribute words
he's, boy, “, ”, tomorrow, he, health, guy, past, moment, i'd, bill, him, king, ...	women's, babe, lady, almost, movie, beauty, bitches, taking, kendall, jenner, wife, close, HUNDRED POINTS SYMBOL, sweetie, ...	flourish, evil, yall, fuckin, slut, females, idiot, niggas, bitches, fuck, bitch, stupid, hate, nigga, ...	thanks, win, long, –, liked, top, star, national, health, 7, public, pretty, amazing, 1st, ...

Table 7: Candidate words extracted from the HSD dataset for gender bias. All capital words represent emojis. We show only the first 14 words here.

Male target words	Female target words	Hateful attribute words	Unhateful attribute words
he's, boy, he, him, king, men's, gentleman, husband	she's, girl, she, her, queen, women's, lady, wife	evil, fuckin, idiot, niggas, fuck, stupid, hate, nigga	thanks, win, long, –, liked, top, star, national

Table 8: Target and attribute words extracted from the HSD dataset for gender bias

Jack Greg Paul Peter Amanda Melanie Kristin Nancy Stephanie Ellen Lauren Emily Rachel

- African American names (target words): Alonzo Jamel Theo Jerome Leroy Torrance Darnell Lamar Lionel Tyree Deion Lamont Malik Terrence Tyrone Ebony Jasmine Tia Yolanda Yvette
- Pleasant attribute words: caress freedom health love peace cheer friend heaven loyal pleasure diamond gentle honest lucky rainbow diploma gift honor miracle sunrise family happy laughter paradise vacation
- Unpleasant attribute words: abuse crash filth murder sickness accident death grief poison stink assault disaster hatred pollute tragedy bomb divorce jail poverty ugly cancer evil kill rotten vomit

B.2 WEAT 4

European and African American names are taken from Bertrand and Mullainathan (2004). As in WEAT 3, we removed some names. Attribute words are the same as WEAT 3.

- European American names (target words): Brendan Greg Brett Jay Allison Anne Carrie Emily Laurie Kristen Meredith Sarah
- African American names (target words): Darnell Hakim Jermaine Kareem Jamal Leroy Rasheed Tyrone Aisha Ebony Keisha Kenya

B.3 WEAT 5

Pleasant and unpleasant attribute words are taken from Nosek et al. (2002a). Target words are the same as WEAT 4.

- Pleasant attribute words: joy love peace wonderful pleasure friend laughter happy
- Unpleasant attribute words: agony terrible horrible nasty evil war awful failure

B.4 WEAT 6

Career and family words are taken from Nosek et al. (2002a). The original WEAT 6 has male and female names as target words. Goldfarb-Tarrant et al. (2021) substituted general male and female words in WEAT 7 for male and female names in WEAT 6 because these names do not appear in WinoBias. Goldfarb-Tarrant et al. (2021) also swapped target and attribute words so that male and female words are attribute words as in WEAT 7 and 8. The word sets used in our experiment are as follows:

White target words	African-American target words	Unhateful attribute words	Hateful attribute words
amazing, automatically, anyone, trecru, giveaway, awesome, nice, seeing, power, here's, pick, easter, aren't, america, virgo, series, photos, less, inside, artist, area, THUMBS UP SIGN, weeks, ...	goin, ion, bad.can, stans, shorty, females, fa, flourish, hoes, niggas, bruh, nigga, yoongi, bout, hoe, nerves, bitches, itgetsworseshow, ain't, FACE PALM OF FEMALE WITH FITZPATRICK SKIN TYPE 4, yall, FACE PALM OF FEMALE WITH FITZPATRICK SKIN TYPE 5, tf, aint, lil, mad, ugly, protesting, mama, sis, tryna, ...	unfollowed, ready, liked, latest, social, –, pretty, excited, light, favorite, far, perfect, easy, public, gemini, following, success, playlist, blue, virgo, ...	slut, bad.can, FACE PALM OF FEMALE WITH FITZPATRICK SKIN TYPE 5, asshole, moron, fucks, females, fa, bitches, bitch, niggas, fuckin, nigga, stupid, SPLASHING SWEAT SYMBOL, flourish, goin, idiot, sh, fucking, stressing, fucked, fuck, itgetsworseshow, ugly, idiots, UNAMUSED FACE, FACE PALM OF FEMALE WITH FITZPATRICK SKIN TYPE 4, bullshit, SMILING FACE WITH HORNS, bastard, shorty, ...

Table 9: Candidate words extracted from the HSD dataset for racial bias

White target words	African-American target words	Unhateful attribute words	Hateful attribute words
amazing, automatically, anyone, awesome, nice, seeing, power, here's, easter, aren't, series, photos, less, THUMBS UP SIGN, weeks	goin, ion, stans, females, fa, bruh, bout, nerves, ain't, yall, aint, lil, mama, sis, tryna	ready, liked, latest, social, –, pretty, excited, light, favorite, far, perfect, easy, public, gemini, following, success, playlist, blue, virgo	slut, asshole, moron, fucks, bitches, bitch, fuckin, stupid, idiot, fucking, stressing, fucked, fuck, ugly, idiots, UNAMUSED FACE, bullshit, bastard, shorty

Table 10: Target and attribute words extracted from the HSD dataset for racial bias

- Career target words: executive management professional corporation salary office business career
- Family target words: home parents children family cousins marriage wedding relatives
- Male attribute words: male man boy brother he him his son
- Female attribute words: female woman girl sister she her hers daughter

B.5 WEAT 7

The word sets are taken from Nosek et al. (2002a). The word "computation" in the math target words does not appear in our processed Tweeter corpus. We remove it in the experiment dealing with Twitter data. To make the target words equal in size, we remove the word "symphony" from the arts target words. ("symphony" was chosen at random.)

- Math target words: math algebra geometry calculus equations computation numbers addition
- Arts target words: poetry art dance literature novel symphony drama sculpture
- Male attribute words: male man boy brother he him his son
- Female attribute words: female woman girl sister she her hers daughter

B.6 WEAT 8

The word sets are taken from Nosek et al. (2002b).

- Science target words: science technology physics chemistry Einstein NASA experiment astronomy
- Arts target words: poetry art Shakespeare dance literature novel symphony drama
- Male attribute words: brother father uncle grandfather son he his him
- Female attribute words: sister mother aunt grandmother daughter she hers her

C Artifacts We Used

Table 11 and 12 show artifacts we used, their creators, versions, and licenses or terms.

D Discussion About the Intended Use of Artifacts We Used

Internet Archive's Terms of Use, which governs usage of the Twitter data, states, "Access to the Archive's Collections ... is granted for scholarship and research purposes only." Our use is in line with its intended use.

E Documentations of the Artifacts

- Wikipedia
  - Language: many languages (We only use the English part.)
  - Type: encyclopedia
  - Demographics of the authors: all over the world
- OntoNotes 5.0
  - Language: English, Mandarin Chinese, Arabic, Chinese (We only use the English part.)
  - Linguistic phenomena: syntactic structure, predicate-argument semantics, word sense disambiguation, coreference, named entity
- WinoBias
  - Language: English
  - Linguistic phenomena: coreference
- Twitter
  - Language: many languages (We only use the English part.)
  - Type: SNS messages
  - Demographics of the authors: all over the world
- The abusive tweet dataset
  - Language: English
  - Type: SNS messages
  - Demographics of the authors: all over the world

F Details of Word Embedding Training

Training settings for word embeddings are similar to Goldfarb-Tarrant et al. (2021). We train Skip-gram word2vec with negative sampling (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017) word embeddings with 300 dimensions. The maximum distance between a word and a predicted

Artifacts	Creators	Versions	Licenses or terms
Bling Fire	Microsoft Corporation (2021)	0.1.8	MIT License
NumPy	Harris et al. (2020)	1.21.6	The 3-Clause BSD License
pandas	pandas development team (2023); Wes McKinney (2010)	2.0.2	The 3-Clause BSD License
SciPy	Virtanen et al. (2020)	1.10.1	The 3-Clause BSD License
`e2e-coref/setup_training.sh`	Lee et al. (2017)		Apache License 2.0
CPython	Python (1990)	2.7.18, 3.7.15, and 3.8.12	Python Software Foundation License Version 2
AllenNLP and AllenNLP Models	Gardner et al. (2018)	2.10.1	Apache License 2.0
PyTorch	Paszke et al. (2019)	2.0.1	The 3-Clause BSD License
TorchText	TorchText maintainers and contributors (2016)	0.15.2	The 3-Clause BSD License
`attract-repel.py`	Mrkšić et al. (2017)		Apache License 2.0
WinoBias	Zhao et al. (2018)		MIT License
The abusive tweet dataset	Founta et al. (2018)	2	Restricted access
`seraphinatarrant/embedding_bias`	Goldfarb-Tarrant et al. (2021)		I got a permission from the corresponding author.
The Word Embeddings Fairness Evaluation framework	Badilla et al. (2020)	0.4.1	MIT License
OntoNotes	Weischedel et al. (2017)	5.0	LDC License
scikit-learn	Pedregosa et al. (2011)	0.24.2	The 3-Clause BSD License
WEAT 3, 4, 5, 6, 7, and 8	Caliskan et al. (2017)		I cited papers from which WEAT words came. (Appendix B)
NLTK	Bird et al. (2009)	3.7	Apache License 2.0
WikiExtractor	Attardi (2015)	3.0.6	GNU Affero General Public License v3.0
Wikipedia	The Wikimedia Foundation (2001)	latest on 2022-10	GFDL and CC BY-SA 3.0
Gensim	Řehůřek and Sojka (2010)	4.2.0	GNU Lesser General Public License v2.1
Matplotlib	Hunter (2007)	3.7.1	LICENSE
ArchiveTeam JSON Download of Twitter Stream 2017-04	Scott (2017)		Internet Archive's Terms of Use
spaCy	Honnibal et al. (2020)	3.7.1	MIT License
The mixed-membership demographic-language model	Blodgett et al. (2016)		MIT License

Table 11: Artifacts we used, their creators, versions, and licenses or terms (Part 1)

Artifacts	Creators	Versions	Licenses or terms
Race and ethnicity data for first, middle, and last names	Rosenman et al. (2023)	9.0	CC0 1.0

Table 12: Artifacts we used, their creators, versions, and licenses or terms (Part 2)

word is set to 5. Word embeddings for coreference resolution are trained on Wikipedia, and for HSD, on the Twitter data. Word embeddings are trained with the NLP library Gensim (Řehůřek and Sojka, 2010).

F.1 Training on Wikipedia

Goldfarb-Tarrant et al. (2021) trained word embeddings on the latest Wikipedia article dump for coreference resolution. Since we cannot identify which version they trained on, we train word embeddings on the latest Wikipedia dump. (The Wikipedia dump was downloaded in October 2022.)

Data preprocessing is almost the same as in Goldfarb-Tarrant et al. (2021). The downloaded Wikipedia article dump is processed with WikiExtractor (Attardi, 2015) to generate a single text file. WikiExtractor outputs documents in the format of

```
<doc id="" url="" title="">
...
</doc>
```

We remove `<doc id="" url="" title="">` and `</doc>` and extract only the contents. Using `text_to_sentences` from Bling Fire (Microsoft Corporation, 2021), we split text into sentences.[5] Tokenization is done by `nltk.tokenize.word_tokenize` from the Python package Natural Language Processing Toolkit (NLTK) (Bird et al., 2009). A word that appears less than or equal to ten times is replaced with a special token `<UNK>`.[6] Unlike Goldfarb-Tarrant et al. (2021), we lowercase the Wikipedia dump after tokenization to reduce the number of vocabulary.

The final Wikipedia dump is the size of 3121412634 word tokens.

[5]The GitHub repository of Goldfarb-Tarrant et al. (2021) mentions that the Wikipedia corpus is preprocessed to have one paragraph per line. We make it have one sentence per line because it is easier to run dataset balancing in this format. The reason we chose `text_to_sentences` is because there is a code using it in Goldfarb-Tarrant et al. (2021)'s repository.

[6]Goldfarb-Tarrant et al. (2021) said they replaced words appearing less than ten times with `<UNK>`, but the code they provided replaces words appearing less than or equal to ten times.

F.2 Training on Twitter

For HSD, Goldfarb-Tarrant et al. (2021) trained word embeddings on 2019 Twitter data. However, tweets in the abusive tweet dataset Founta et al. (2018) were collected from 2017-03-30 to 2017-04-09. We train word embeddings on Twitter data from 2017-04-01 to 2017-04-06 to perform better on HSD. We train them on the data from 04-01 because the Internet Archive, from which we downloaded Twitter data, offers Twitter data per month. We use the data until 04-06 to make the data size on par with the data used in Goldfarb-Tarrant et al. (2021).

Data preprocessing is almost the same as in Goldfarb-Tarrant et al. (2021). They removed retweets from the tweet dataset. The hashtag symbol (#), @-mention, and URL are replaced with special tokens `<HASH>`, `<MENTION>`, and `<URL>`, respectively. Tweets are tokenized by `nltk.tokenize.TweetTokenizer` from NLTK (Bird et al., 2009). Tokens appearing less than ten times are replaced with `<UNK>`. The difference we make in preprocessing is that we remove tweets withheld in some countries and ones withheld due to DMCA (the Digital Millennium Copyright Act) complaints, as well as retweets. The final Twitter data is the size of 42429319 word tokens.

G The Number of Parameters in the Models Used, the Computational Budget, and Computing Infrastructure Used

The number of trainable parameters in the coreference resolution system is 1757195. The number of trainable parameters in the HSD system is 360902.

Here, we only explain computationally heavy operations in our study.

Extracting text data from the Wikipedia article dump takes about 4 hours with 112 CPUs (Intel Xeon Gold 6258R).

Training one word2vec on the Wikipedia article dump takes about 4 hours with 9 CPUs (Intel Xeon Platinum 8360Y). Training one fastText on the Wikipedia article dump takes about 8 hours

Dataset	Number of examples		
	Train	Val	Test
OntoNotes 5.0	75187	9603	9479
WinoBias Type 1		792	792
WinoBias Type 2		792	792
The HSD dataset	76496	8500	15000

Table 13: The number of examples in each dataset. An "example" here means a tweet for the HSD dataset and a sentence for the others.

Male-targeting	Female-targeting	Neutral
1837	1594	11569

Table 14: The number of male-targeting, female-targeting, and neutral tweets in the test set of the HSD dataset.

with 9 CPUs (Intel Xeon Platinum 8360Y). We train 90 word embeddings for each embedding algorithm. In total, training word embeddings on the Wikipedia article dump takes about 1080 hours with 9 CPUs.

Expanding word sets takes around 3 hours with 9 CPUs (Intel Xeon Platinum 8360Y).

Dataset balancing takes around 7.5 hours with 36 CPUs (Intel Xeon Platinum 8360Y). ATTRACT-REPEL takes around 9 hours with 8 GPUs. Training one coreference resolution model takes about 1.25 hours with 1 GPU. We train 216 models.

Measuring WEAT takes around 2 hours with 9 CPUs (Intel Xeon Platinum 8360Y). Evaluating coreference resolution models on WinoBias and the test data of CoNLL-2012 Shared Task (Pradhan et al., 2012) takes around 2 hours with 9 CPUs (Intel Xeon Platinum 8360Y).

The GPUs we use are NVIDIA A100.

H Basic Statistics of the Datasets for Extrinsic Bias Metrics

Basic statistics of the datasets for extrinsic bias metrics are shown in Table 13, 14, and 15.

White-sounding	African-American-sounding	Others
9722	1798	3480

Table 15: The number of white-sounding, African-American-sounding, and the other tweets in the test set of the HSD dataset.

I Bias Modification Word Sets

Goldfarb-Tarrant et al. (2021) used `en_vectors_web_lg` from spaCy (Honnibal et al., 2020) as v. We use `en_core_web_lg` since we cannot download `en_vectors_web_lg`. Goldfarb-Tarrant et al. (2021) used $K = 100$, but we use $K = 10$ to prevent inappropriate words sneaking in.[7] We remove odd terms from the expanded word sets. Goldfarb-Tarrant et al. (2021) does not mention what constitutes "odd." In our experiment, for white-sounding and African-American-sounding word sets, we use the mixed-membership demographic-language model developed by Blodgett et al. (2016) to rule out odd terms. For European-American and African-American names word sets, we use a dataset by Rosenman et al. (2023), containing the first names' race and ethnicity probabilities. For 136K first names, it provides racial and ethnic probabilities of white, black, Hispanic, Asian, and others. By taking argmax, we estimate races and ethnicities of expanded European-American and African-American names and remove inappropriate names from the word sets. We manually decided whether to remove words (or names) not in the demographic-language model (or the name dataset) and words in the other word sets. We did this removal, considering whether words are unnatural to be in the word set. (E.g., "kid" is unnatural to be in a male word set, and "purple" is in a pleasant word set.) Final expanded word sets are accessible in the supplementary material.

J Acknowledgement of AI Use

We used Code Llama as a coding assistant. We also used Google Bard to ask questions related to coding.

K Measuring the RNSB

Our implementation of RNSB is based on an implementation in the Word Embeddings Fairness Evaluation framework (WEFE) (Badilla et al., 2020). Logistic regression of the RNSB is `sklearn.linear_model.LogisticRegression` from scikit-learn (Pedregosa et al., 2011). Arguments to `sklearn.linear_model.LogisticRegression` is in Table 16. By default, the RNSB in WEFE

[7] For example, we got "girly" for male attribute expansion when $K = 100$.

Argument	Value
solver	"liblinear"
max_iter	10000

Table 16: Arguments to `sklearn.linear_model.LogisticRegression`. The other arguments are set to default values.

holds out 20% of attribute words as test data. We disable this behavior so that logistic regression is trained on full attribute words. We train logistic regression ten times with different random seeds. The averaged RNSB is used to calculate correlations with extrinsic bias metrics.

L Hyperparameter Values of Hate Speech Detector

We use the same hyperparameter values as in Kim (2014). However, Kim (2014) did not mention the mini-batch sizes of validation and test datasets. We tested the mini-batch sizes of 1 and 50 (the mini-batch size of a training dataset) and found that an F1 score on the test set is higher with the mini-batch size of 50. Therefore, we set 50 as the mini-batch sizes.

M Preprocessing of the Abusive Tweet Dataset

The abusive tweet dataset (Founta et al., 2018) labeled each tweet with one of the following four labels: abusive, hateful, spam, and normal. Referring to the annotation scheme of Shared Task "SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter" (Basile et al., 2019), we convert the dataset to an HSD dataset by translating abusive and hateful labels to hate speech labels and spam and normal labels to non-hate speech labels. We select around 10% of tweets from non-test data as validation data, following Kim (2014).

Goldfarb-Tarrant et al. (2021) did not mention the preprocessing of the abusive tweet dataset. We investigated the preprocessed data they shared and tried to infer how the preprocessing was done. We preprocess the abusive tweet dataset like we do the Twitter data for word embedding training, although we do not filter out tweets. Words not in the word embedding vocabulary are replaced with `<UNK>`, except ones that appear more than or equal to ten times in the training section of the abusive tweet dataset.

N Modifications to Packages

N.1 WEFE

- Fix comments and error messages
- Handle multiple token expression
- When measuring the RNSB, the original WEFE raises an exception if the number of positive training examples is 1. Probably, this is a mistake for "if the number of positive training examples is less than 1." We fixed it.
- Fix a docstring
- Fix WEAT words
- Implement an extension of the RNSB

N.2 AllenNLP and AllenNLP Models

When measuring the WinoBias score, we use modified versions of AllenNLP and AllenNLP Models. We relaxed the specifications in the requirement files so that the packages are installed successfully. We modified AllenNLP Models to handle WinoBias.

N.3 The Mixed-Membership Demographic-Language Model

We adapted the mixed-membership demographic-language model by Blodgett et al. (2016) to Python 3.

N.4 WikiExtractor

When we ran WikiExtractor, an exception was raised. We searched GitHub issues for the same problem and applied a fix found there.

O The Combinations of Parameters of ATTRACT-REPEL

ATTRACT-REPEL (Mrkšić et al., 2017) has following parameters: δ_{sim}, δ_{ant} ("the similarity and antonymy margins"), λ (regularization coefficient), k'_1, and k'_2 (mini-batch sizes for synonym and antonym pairs). Mrkšić et al. (2017) conducted a parameter search in the space of $\delta_{\text{sim}}, \delta_{\text{ant}} \in \{0, 0.1, 0.2, ..., 1\}$, $\lambda \in \{10^{-3}, ..., 10^{-10}\}$, and $k'_1, k'_2 \in \{10, 25, 50, 100, 200\}$. They reported that $\delta_{\text{sim}} = 0.6$, $\delta_{\text{ant}} = 0$, $\lambda = 10^{-9}$, and $k'_1 = k'_2 = 10, 25, 50$ achived the best result.

When Goldfarb-Tarrant et al. (2021) modified word embeddings, they used the best parameter setting with $k'_1 = k'_2 = 50$.

Taking those into account, we chose the following combinations of the parameters: $\delta_{\text{sim}} \in \{0, 1\}$, $\delta_{\text{ant}} \in \{0, 1\}$, $\lambda \in \{10^{-1}, 5 \times 10^{-2}, 10^{-2}\}$, and $k'_1, k'_2 = 50$.[8]

P Dataset Anonymization / Offensive Content Removal

When preprocessing datasets, we mostly follow Goldfarb-Tarrant et al. (2021). To make our experiment close to theirs, we do not do additional dataset anonymization or offensive content removal.

P.1 Twitter Data

An @-mention is replaced with a special token `<MENTION>`.

P.2 The Abusive Tweet Dataset

The abusive tweet dataset from Founta et al. (2018) contains offensive content. Since we train hate speech detectors, we do not filter out offensive content.

An @-mention is replaced with a special token `<MENTION>`.

[8] We found that $\lambda = 10^{-3}$ did not yield much debiasing (and over-biasing) effect compared to $\lambda = 10^{-2}$. Therefore, we only consider $\lambda \geq 10^{-2}$.