aff#9

Debiasing Word Embeddings from Sentiment Associations in Names

Christoph Hube, Maximilian Idahl, and Besnik Fetahu

L3S Research Center, Leibniz University of Hannover
Hannover, Germany
{hube, idahl, fetahu}@L3S.de

ABSTRACT

Word embeddings trained through models like skip-gram, have shown to be prone to capturing the *biases* from the training corpus, e.g. *gender bias*. Such biases are unwanted as they spill in downstream tasks, thus, leading to discriminatory behavior.

In this work, we address the problem of *prior sentiment* associated with *names* in word embeddings where for a given *name* representation (e.g. "*Smith*"), a sentiment classifier will categorize it as either *positive* or *negative*. We propose *DebiasEmb*, a skip-gram based word embedding approach that, for a given *oracle* sentiment classification model, will debias the name representations, such that they cannot be associated with either *positive* or *negative* sentiment. Evaluation on standard word embedding benchmarks and a downstream analysis show that our approach is able to maintain a high quality of embeddings and at the same time mitigate sentiment bias in name embeddings.

ACM Reference Format:
Christoph Hube, Maximilian Idahl, and Besnik Fetahu. 2020. Debiasing Word Embeddings from Sentiment Associations in Names. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* WSDM, Houston, Texas, USA, 9 pages. https://doi.org/10.1145/3336191.3371779

1 INTRODUCTION

Word embeddings are one of the most basic representations of words in natural language understanding. Their use in downstream tasks has shown great benefits on a variety of tasks, such as named entity recognition and part of speech tagging [22, 24] based on their ability to capture the word context.

Due to the way embeddings are trained, they often have been shown to contain biases, e.g. *gender bias*. These biases are reflected in terms of words that are supposedly to be either *gender neutral* or any other form of *word categorization*, but instead are shown to be in close proximity to words that belong to explicit categories, such as *gender*. Research has shown that specific job roles reflect stereotypes of a specific culture or group [3, 4], e.g. *"nurse"* being closer to *"female"* and *"programmer"* being closer to *"man"*, etc.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371779

Such findings concur with sociolinguistic theory [10], which states that language and its structures is a medium that is in the function of its *social groups*.

In the case of word embeddings the biases stem from the underlying training corpus. Even for large corpora, approaches such as word2vec [22] trained on Google News, and GloVe [24] trained on Wikipedia, exhibit bias regarding gender and race.

Current state of the art approaches aim to debias embeddings by removing the direction of the *protected attribute* (e.g. gender), such that a target word is *equidistant* to all categories of the protected attribute (e.g. *gender roles*)[1]. This type of intervention is done either as a post-processing step [3], pre-processing step [5], or directly during training [31]. However, analysis [9] has shown that these approaches mitigate bias only at a superficial level, where much of the initial bias can be recovered through *word proxies*, words that are in close proximity in the vector space to the words being used for bias mitigation.

In this work, we tackle the problem of *sentiment association* with *names* in word embeddings. That is, given a training corpus, some names may be co-occurring more frequently with words of either positive or negative sentiment. Hence, according to the distributional hypothesis [11], the embeddings of such names will be close to the words with explicit sentiment with which they co-occur. For instance, Caliskan et al. [4] shows that *European American* first names are associated with pleasant words, while *African American* names are comparably stronger related with unpleasant words. Furthermore, *male* first names are associated with career words, whereas *female* names are stronger connected to family words. Similarly, we show that last names suffer from similar biases, and that these biases can heavily impact the decision-making of downstream models such as sentiment classifiers [5, 17].

For example, if we are given two *unnuanced factual* sentences as the ones shown below, a reliable sentiment classifier that uses embeddings as a means to represent the words in a sentence should classify both statements as *neutral*.

- *"**Obama** is president."*
- *"**Trump** is president."*

While for specific tasks, sentiment association of names may be essential, in the case of pre-trained word embeddings, where a name may correspond to multiple real-world persons, they are unsuitable as they result in significant discrimination and other forms of biases.

We propose *DebiasEmb*, a novel approach that debiases word embeddings from sentiment association in names during training. During the training phase of the skip-gram model, apart from the

[1] In most cases this is done at a binary level, e.g. *male* and *female* for genders.

objective of predicting the *context word*, through an *oracle classifier* we additionally ensure that the *center words* of interest cannot be associated with neither positive or negative sentiment. The oracle classifier in this case is a pre-trained sentiment classification model with positive and negative words, respectively from their embeddings.

The novelty of DebiasEmb is that it seamlessly integrates the standard word embedding objectives together with any debiasing component, such as debiasing sentiment or other word categories. To show the effectiveness of our approach, we evaluate DebiasEmb on two sets of tasks. First, at word level, we ensure that the names from a given name list cannot be associated with neither the positive nor negative sentiment class and that the constructed embeddings do not suffer in terms of quality when compared to the original skip-gram embeddings. Second, on a downstream task we show the debiasing effect of DebiasEmb on a text-level sentiment classifier. In summary, our contributions are the following[2]:

- DebiasEmb, a novel approach for debiasing word embeddings from sentiment association in names;
- Thorough evaluation of the approach, including an extrinsic downstream analysis based on a sentiment classifier trained on two different datasets (reviews and news data).

2 RELATED WORK

Research on bias in word embeddings focuses strongly on gender and racial bias. The seminal work by Bolukbasi et al. [3] shows how a gender direction in the vector space of word embeddings can be defined by using pairs of gender specific words, namely feminine and masculine words (*she-he* pairs). They point out that many gender-neutral words are associated with one gender, e.g. *doctor* is strongly shifted towards male, while *nurse* is associated with the female gender, reflecting societal gender stereotypes, even when using popular training datasets such as Google News articles. To mitigate gender bias in word embeddings they introduce a post-processing approach that removes the identified gender direction from a pre-defined list of words, while keeping it for words that convey an explicit gender function (e.g. *mother*, *father*, *boy*, *girl*). This approach has been criticized for being "fairness through blindness", i.e. removing relevant information while not covering important bias aspects such as proxies [4], and for relying on a classifier to identify the definition of words, which could lead to errors being propagated into the model [31].

Zhao et al. [31] aim at isolating the gender attribute into one component of the resulting word vectors, while removing it from all other components. This is achieved during training by modifying the loss function and using a list of gender seed words.

However, a recent study by Gonen and Goldberg [9] shows that both, the post-processing and the isolation approach, remove bias only at a superficial level. The gender bias is still present in the resulting embeddings. An interesting observation in this case is that the gender direction, that is limited to a specific word list, does not cover sufficiently other word proxies that may introduce bias. Hence, this should be used with precaution and can serve only as an indicator for bias. On the other hand, the debiasing approach that is done during the training phase [31] is preferable over post-processing. However, even in this case, the limitation is in defining genders, namely words that are specific for a gender. This is similar to the work by Bolukbasi et al. [3].

Contrary to the previously described works, our approach has the advantage of using a pre-defined classification model (i.e. the oracle sentiment classifier) that is trained on a specific seed set of words, respectively their embeddings, with explicit sentiment. Consequentially, this allows us to address word proxies that may associate names with prior sentiment, or other biases for other word categories like gender, race etc., through the similarities in the vector space of the set of seed words with other proxy words.

Caliskan et al. [4] introduce the Word Embedding Association Test (WEAT), an intrinsic test for measuring biases in word embeddings. It determines the mean proximity of words in two target groups to words in two attribute groups (e.g. *Pleasant*, *Unpleasant*). Swinger et al. [28] propose an unsupervised algorithm for automatically outputting WEAT tests that does not require the sensitive group (e.g. gender, race) to be specified. They find gender biases even for word embeddings that have been debiased using the approach introduced by [3]. WEAT has recently been extended for measuring bias in sentence encoders [21].

Contrary to [4, 28] we do not measure bias intrinsically, but extrinsically on downstream tasks to observe the actual impact that the identified biases have on the behavior of downstream models. We also propose an approach for debiasing embeddings.

In their analysis, Diaz et al. [5] find significant age-related bias in a variety of sentiment analysis models and popular GloVe word embeddings. They introduce a simplistic approach for debiasing by removing all occurrences of the protected attribute (i.e. age) from the input data. In contrast, our approach does not modify the input data but instead debiases embeddings during training.

Recent research has introduced a new generation of contextualized word embeddings that are able to represent polysemy. [30] show that contextualized word embeddings such as ELMo [25] show similar bias compared to common word embeddings such as GloVe. They find gender bias in the trained embeddings intrinsically and mitigate bias on the downstream task of coreference resolution by leveraging augmented training data with swapped gender words during training or post-processing. A shortcoming of this approach is that it only applies to gender, since in other contexts (e.g. race, sentiment), a clear opposite word can not be defined. In contrast, our approach is domain-independent and applicable to all types of class-based biases.

Another line of work aims to detect biased language in text directly using feature-based [12, 26] or neural-based approaches [13, 14]. Diaz et al. [5] uses heuristics to identify all occurrences of a protected attribute. These approaches could be used to remove biased statements from the training data, though it is not clear whether this would cover all types of biases, especially the ones that are introduced through proxies. Using a pre-processing approach that removes entire statements containing biased language would also lead to a situation where valuable parts of the training data co-occurring with biased parts would be lost as well a parts being misclassified as bias. Our approach does not remove training data but instead debiases the resulting embeddings during training.

[2]Code and data used in this work have been published at https://github.com/ChristophHubeL3S/debiasEmb

3 DEBIASED WORD EMBEDDINGS

In this section, we describe our approach *DebiasEmb* for training debiased word embeddings. *DebiasEmb* consists of two main components: (i) an oracle sentiment classifier, and (ii) the modified skip-gram with negative sampling (SGNS) model. In the following we explain in details the individual components.

3.1 Oracle Sentiment Classifier

To determine if names have a *prior sentiment bias* in word embeddings, we use pre-trained supervised models, trained on embeddings from words that contain *explicit* prior sentiment, e.g. lexicons of *positive* and *negative* sentiment from SentiWordNet [1] or other hand-crafted lexicons [18]. More specifically, for any other words, e.g. *proper nouns*, if such a model is able to classify them as either "*positive*" or *"negative"*, that is an indicator of bias in the corresponding word embedding. For example, *"Smith"*, *"Li"*, or *"Mohamed"* should not be associated with any prior sentiment.

In this work, we consider the oracle classifier to be a pre-trained *logistic regression* model (see Equation 1), which for each embedding dimension associates a feature weight. However, any classification model that uses a differentiable classification function can be used in this case.

$$f_{LR}(\mathbf{x}) = \sigma\left(\frac{1}{1 + e^{-(\mathbf{w}\cdot\mathbf{x}+b)}}\right) \tag{1}$$

where, $\mathbf{x}$ represents the embedding of a specific word, and $\mathbf{w}$ represents the weights associated with each embedding dimension.

In an ideal scenario, the classifier outputs $f_{LR} = 0.5$, which results in the *inability* of the classifier to predict the sentiment of a word from the protected class. In the next section, we show how we debias word embeddings given some pre-trained sentiment classifier. We will use the classification model to guide the debiasing process of word embeddings such that target names cannot be associated with any sentiment score.

3.2 Debiased Word Embedding Model

The main intuition behind the skip-gram with negative sampling model [23] is to use the *context* of a *center* word to learn its representation. The objective function of the SGNS model in Equation 2 is to maximize the similarity of the center and context words, while at the same time minimize the similarity of the center word against non-context words (negative samples).

$$f_{SGNS} = \frac{1}{N}\sum_{i=1}^{N}\sum_{-c\leq j\leq c}\log p(w_{i+j}|w_i) \tag{2}$$

where, c is the context window of the center word. The above function shows the ability of the model to predict the context word w_{i+j} given the center word w_i. The training is done by minimizing the following loss function $\mathcal{L}$.

$$\mathcal{L}_{sgns} = \log\sigma(\mathbf{v}_{\mathbf{w_j}}\cdot\mathbf{u}_{\mathbf{w_i}}) + \sum_{k=i}^{K}\mathbb{E}_{w_k\sim P(w_k)}\log\sigma(-\mathbf{v}_{\mathbf{w_k}}\cdot\mathbf{u}_{\mathbf{w_i}}) \tag{3}$$

Despite the ability of these models to efficiently capture the word meaning based on its context, there are several documented issues that the embedding models capture, such as *societal biases* (i.e. gender, racial biases) and other issues that are encoded in the textual resources, from which the training data are drawn [3, 6, 30].

We propose a modified version of the SGNS model, where we modify the loss function such that the computed word embeddings for a target set of words in the vocabulary, e.g. names are not associated with any sentiment score.

$$\mathcal{L}_{sent} = \left|\sigma\left(\mathbf{w}\cdot\mathbf{u}_{w_i}\right) - \frac{1}{2}\right| \tag{4}$$

where, $\sigma(\cdot)$ represents the pre-trained oracle classifier, respectively, we use the weights associated with the embedding dimensions and assess whether the word embedding $\mathbf{u}_{w_i}$ of our center word w_i encodes any prior sentiment bias. For a word embedding to be bias free, the classifier should not be able to distinguish between the *positive* or *negative* sentiment categories, thus, the value of the σ function will be equal to 0.5, which will result in zero loss. In the other cases, we aim at changing the embeddings of a center word $\mathbf{u}_{w_i}$ such that the distance in $\mathcal{L}_{sent}$ is minimized.

Finally, the loss function of *DebiasEmb* is the combined sum of the original loss function of the SGNS model and the loss function that measures the sentiment bias score.

$$\mathcal{L} = \mathcal{L}_{sgns} + \mathcal{L}_{sent} \tag{5}$$

Note that, our model can incorporate any oracle classifier whose classification function is differentiable, and additionally, we are not limited to only sentiment bias.

4 WORD EMBEDDING EXPERIMENTAL SETUP

In this section, we describe the evaluation setup for our approach *DebiasEmb*. Namely, we evaluate its capability to reduce sentiment bias association with names. Additionally, we introduce competitors against which we compare and the evaluation metrics to measure sentiment bias in word embeddings.

4.1 Datasets for Training Word Embeddings

Word embeddings trained on news datasets have been widely used for a span of different tasks, including sentiment analysis [8], part-of-speech tagging [29], and named entity recognition [27]. For the evaluation of *DebiasEmb*, we use six different news datasets with randomly selected sentences from popular news sources in the time period of 2013-2017. The specifics of each dataset are shown in Table 1. The *Random* dataset contains sentences that were randomly selected from a set of more than 100 different news sources.

	#Sentences
HuffingtonPost	4,631,874
Random	4,477,326
Breitbart	1,333,510
CNN	1,252,718
BBC	773,657
RussiaToday	457,487

Table 1: Number of sentences for each news dataset.

4.2 Name List

We extract an extensive list of 130,000 surnames that are used in the United States of America from Mongabay[3]. From this list, we filter out all names that occur less than 10 times in our combined news dataset, as well as all names that occur more often in lower case than in uppercase (first letter of the word is capital), indicating that these names have ambiguous meaning, and do not serve solely the functions of proper nouns (e.g. *Bottom*, *Speech*). This results in 17,055 names for which we aim to debias the word embeddings.

4.3 Baselines for Debiasing Word Embeddings

We compare DebiasEmb against the following baselines:

- **SGNS:** Default SkipGram model with negative sampling, as introduced in [23], no debiasing.
- **PostDebiasing:** Debiasing approach introduced in [3]. Instead of the *gender direction* (*he - she*), we use a *sentiment direction* (*positive - negative*). We replace the definition of word pairs with a set of positive-negative pairs, e.g. "great" - "terrible", "positive" - "negative", "competent" - "incompetent". We first train the word embeddings using SGNS and then remove the sentiment direction from all name words by applying the post-processing step.
- **PreDebiasing:** Debiasing approach introduced in [5]. The idea is to remove all occurrences of the protected attribute from the input data. Instead of sentences containing age words, we remove all sentences containing at least one name from the name list *and* at least one positive or negative word from our word list as introduced in section 3.1. Note that simply removing all sentences containing a name from the name list (even the ones without a positive or negative word) is not an option since in this case the resulting word embeddings would not contain the name at all and all information about this name would be lost, even the non-biased information. This would be fairness through blindness.
- **PreDebiasEmb:** Combination of PreDebiasing and DebiasEmb. We first apply PreDebiasing on the input data and then use DebiasEmb during training.

We do not compare against the approach introduced by [31] since this approach is based on the same definition of the protected attribute as the approach in [3]. Focusing the sentiment direction into one component of the resulting vectors and then removing this component is conceptionally the same approach as instantly removing the sentiment direction from the embeddings.

4.4 Word-Level Sentiment Classifier

For the evaluation on word-level, we use a classifier that is identical with the oracle sentiment classifier, as introduced in section 3.1. Based on the embedding of the input word, it outputs the class probability for both classes. Due to the classification task being binary and the probabilities summing up to 1, it is sufficient to focus on the probability of the positive class. Its score is between 0 (negative) and 1 (positive). Both, the oracle and the evaluation classifier have been trained on a word lexicon containing 2004 positive and 4782 negative words [18].

[3]https://names.mongabay.com/data/surnames_A.htm

4.5 Bias Measures

To evaluate the amount of *sentiment bias* associated with names for some given embeddings, we consider the following measures that use the *word-level sentiment classifier* to obtain the classification label and class probability scores for a given name.

Dist: Here we measure the ability, respectively, the inability of a classification model $\sigma(\cdot)$ to categorize a name as either having *positive* or *negative* sentiment. For a binary classification model, a class probability of 0.5 results in the model's inability to categorize the name into either of the sentiment categories. Thus, the smaller the distance to 0.5 the lower the bias. For a given set of names N, we formalize **Dist** as the mean score across all names.

$$\textbf{Dist} = \frac{1}{|N|} \sum_{n \in N} |\sigma(\mathbf{w} \cdot \mathbf{u}_n) - 0.5| \tag{6}$$

Var: In addition to **Dist** we also take into account the variance of classification scores $\sigma(\cdot)$ across names. Through this measure we aim at capturing if the produced embeddings have varying bias behavior across the different names. For instance, if *all names* are categorized with the *same* sentiment category, e.g. *positive sentiment* with high class probability of 0.9, consequentially, the embeddings do not discriminate against specific names, however, they contain positive bias towards names with **Dist**=0.4. On the contrary, if for specific names the classification model $\sigma(\cdot)$ yields varying sentiment categories with varying probability scores, then, the resulting embeddings discriminate against specific names. Thus, values that are zero or close to zero, indicate bias free embeddings.

5 WORD-LEVEL EVALUATION RESULTS

In this section, we show the evaluation results at the word level. First, we report the results in terms of bias for the different competing approaches. Second, we report the quality of the computed embeddings, computed on standard benchamarking datasets. Finally, we show a detailed analysis of names associated with positive/negative words and their proxies.

5.1 Embedding Debiasing Results

Debiasing Results. Table 2 shows the results for the **Dist** and **Var** measures based on the word-level sentiment classifier for all datasets and all approaches. We observe a debiasing effect of DebiasEmb for all news datasets. On average, DebiasEmb outperforms all baselines in terms of **Var** with a relative improvement of 86% when compared to SGNS. It also achieves a relative improvement of 52% for **Dist** compared to SGNS, showing that DebiasEmb not only debiases the embeddings, resulting in the inability of the word-level sentiment classifier in classifying names, but it additionally increases the homogeneity of the class probabilities across names and therefore decreases the name bias of the classifier.

On the other hand, the PostDebiasing baseline does not show significant improvement over SGNS. Contrary, the PreDebiasing approach performs well, especially for the **Dist** measure. However, when combining the PreDebiasing and DebiasEmb, we achieve the lowest **Dist** value with a relative improvement of 65% compared to SGNS and a relative improvement of 78% in terms of **Var**. This shows that PreDebiasing is another effective approach for debiasing embeddings, especially when combined with DebiasEmb. The results are consistent across all news datasets that were used for training embeddings, with only minor differences.

	SGNS		DebiasEmb		PostDebiasing		PreDebiasing		PreDebiasEmb	
	Dist	Var	Dist	Var	Dist	Var	Dist	Var	Dist	Var
BBC	0.246	0.0554	0.151	0.0141	0.344	0.0206	0.093	0.0158	0.073	0.0109
Breitbart	0.211	0.0413	0.087	0.0043	0.197	0.0383	0.082	0.0167	0.034	0.0031
CNN	0.214	0.0164	0.160	0.0005	0.274	0.0124	0.102	0.0175	0.087	0.0048
HuffingtonPost	0.214	0.0210	0.114	0.0012	0.161	0.0300	0.102	0.0208	0.121	0.0108
RussiaToday	0.194	0.0103	0.028	0.0001	0.137	0.0207	0.068	0.0103	0.039	0.0060
Random	0.227	0.0554	0.086	0.0088	0.320	0.0280	0.098	0.0159	0.104	0.0080
Mean	0.218	0.0333	0.104	**0.0048**	0.239	0.0250	0.091	0.0162	**0.076**	0.0073

Table 2: Mean distance from 0.5 (Dist) and variance (Var) for name words using the *word-level* sentiment classifier. Results are shown for all approaches and all datasets. Lowest values for Dist and Var are highlighted.

	Accuracy
SGNS	0.77
DebiasEmb	0.72
PostDebiasing	0.78
PreDebiasing	0.70
PreDebiasEmb	0.69

Table 3: Average word-level model accuracy for all approaches.

	SGNS	DebiasEmb	PreDebiasing	PreDebiasEmb
AP	0.286	0.289	0.268	0.255
BLESS	0.325	0.324	0.307	0.300
Battig	0.178	0.180	0.157	0.155
ESSLI_1a	0.489	0.473	0.511	0.477
ESSLI_2b	0.650	0.621	0.700	0.758
ESSLI_2c	0.478	0.478	0.485	0.478
MEN	0.194	0.192	0.202	0.204
MTurk	0.296	0.293	0.300	0.297
RG65	0.153	0.154	0.085	0.073
RW	0.084	0.080	0.187	0.185
SimLex999	0.058	0.058	0.088	0.093
TR9856	0.101	0.102	0.112	0.114
WS353	0.162	0.171	0.220	0.231
WS353R	0.198	0.206	0.198	0.215
WS353S	0.194	0.203	0.295	0.309
Google	0.076	0.075	0.043	0.044
MSR	0.131	0.130	0.070	0.071
SemEval2012_2	0.088	0.086	0.092	0.092

Table 4: Performance on benchmarks for word embeddings. The results show that debiasing efforts do not have significant negative impact on the performance of the embeddings.

In summary, the results from Table 2 show that sentiment bias associated with names is present in news corpora. Approaches for training embeddings, like SGNS, associate names with sentiment categories, a consequence of the training corpora, where names co-occur with words that have explicit sentiment valence. These associations can not be remedied by simply relying on specific lexicons for debiasing as in the case of PostDebiasing. Our approach through an oracle classifier can guide the computation of embeddings such that, apart from capturing word meaning following the distributional hypothesis [11], it additionally constrains the parameters of the embeddings and does not allow names from a given target set to be associated with any sentiment category.

Word-level Classifier Accuracy. Table 3 shows the word-level sentiment classification accuracy during training with the corresponding embeddings of the words in the lexicon containing positive and negative words (cf. Section 4.4). The scores are shown for all five approaches, respectively the produced embeddings, averaged across all news datasets.

In terms of accuracy, SGNS and PostDebiasing achieve the best performance. DebiasEmb and the PreDebiasing combination achieve slightly lower performance. This shows that there is a slight trade-off in terms of debiasing embeddings and correspondingly the ability of the word-level classifier to distinguish between positive and negative words. While a high classification score correlates with the reliability of the bias measures, in the following sections we show that in terms of embedding quality in standard benchmarks, the embeddings computed through DebiasEmb have only very minor differences. And as we will show in the downstream task evaluation in Section 6, where we train a text-level sentiment classifier, models that represent the word using DebiasEmb embeddings achieve significantly lower bias in determining the sentiment of a sentence, that is, with varying names the sentiment of the sentence does not change.

5.2 Benchmark Testing

To ensure that the quality of the resulting embeddings does not suffer due to the debiasing efforts, we compare the computed embeddings based on our DebiasEmb approach, and other competitors, against the standard SGNS embeddings on standard benchmark tests [15]. It is important to note here that due to the limited genre and scope of news corpora, the scores on certain benchmarks may be lower when compared to embeddings trained on more generic corpora like Wikipedia. Hence, we use the SGNS embeddings as a reference point for comparison.

Table 4 show the results for the different types of embeddings. Each value is computed as the mean over all the embeddings from all the different news sources. We note that there is no significant difference between the different embeddings. Thus, concluding that such debiasing efforts do not harm the resulting quality of embeddings.

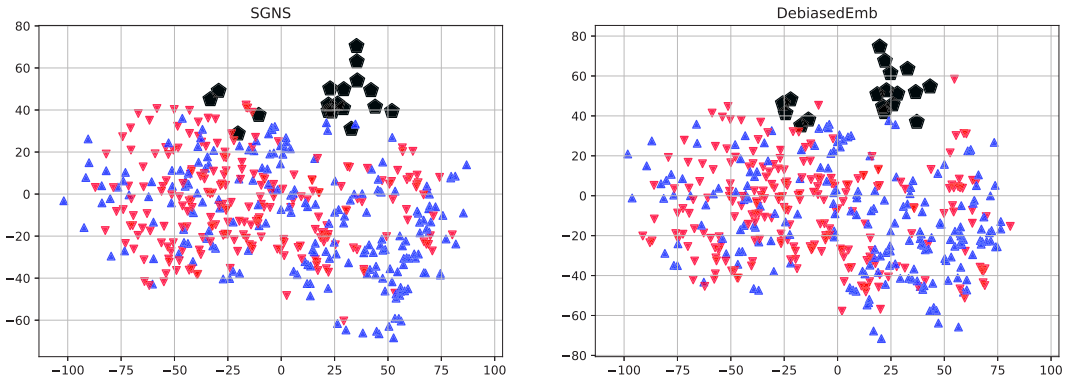

Figure 1: Position of word vectors using t-SNE and the two most important components. Black pentagons represent politician names, blue triangles positive words, and red triangles negative words. Names have a farther distance to the positive and negative words in the case of *debiased* word embeddings compared to *SGNS* word embeddings, as measured based on the *Euclidean distance*.

5.3 Arrangement of Names in Vector Space

Finally, Figure 1 shows the projection of the name embeddings[4] based on the t-SNE [20] non-linear dimensionality reduction technique. The projection on the two most important components reveals that the names from DebiasEmb are more closely clustered together, and furthermore are more distant to words with explicit sentiment and their close proxies. On the positive dimension, the Euclidean distance in the case of DebiasEmb is $DebiasEmb_{POS}$ = 92.71, contrary to $SGNS_{POS}$ = 79.83. Similarly, on the negative dimension the Euclidean distance in the case of DebiasEmb is $DebiasEmb_{NEG}$ = 91.72, contrary to $SGNS_{NEG}$ = 79.12. The Euclidean distance confirms the results we achieve in Table 2.

6 DOWNSTREAM ANALYSIS SETUP

Word embeddings are rarely leveraged for word-level tasks, such as predicting the sentiment of a single word, more commonly they are applied to sentences or longer textual snippets.

In this section, we introduce the setup for applying the trained embeddings using SGNS and DebiasEmb to a downstream classifier, which predicts the sentiment of a textual snippet containing a name (e.g. movie reviews or quoted statements taken from news articles). We analyse how the encoded name biases in embeddings are spilled onto downstream models, and how using the debiased embeddings changes the behavior of the model.

6.1 Classifier Training Data

We use two datasets for labelling text snippets with their sentiment label. The first dataset contains movie reviews and is used widely on this particular task. Whereas, the second contains sentences from news articles, and was created such that the *genre* is the same as the training corpora of the trained embeddings. Both datasets contain textual snippets only in the English language.

- **Reviews:** This dataset contains 12,500 positive and 12,500 negative movie reviews extracted from the Internet Movie Data Base (IMDB) and is openly available [19].
- **News:** Since there is no openly available news dataset for sentiment analysis that is suitable for our problem setting, we constructed a news dataset and labeled it by means of crowdsourcing. Similarly to Balahur et al. [2], we extracted a sample of quotation phrases from our news collection containing at *least one name* and let crowdworkers decide if the phrases are positive, negative, or neutral, using majority voting with three judgments per phrase. The agreement rate, as measured through Fleiss' Kappa, is κ = 0.329. We discard phrases where no agreement could be reached. The final news dataset contains 1804 positive and 2402 negative sentences. We additionally discard neutral sentences, as we are focusing on binary sentiment classification in this work.

Table 5 provides a summary of the datasets. In the **Reviews** dataset, 21,861 out of 25,000 instances contain at least one person name, extracted through Named Entity Recognizer (NER) [16].

	pos	neg	total	cont. names
Reviews	12,500	12,500	25,000	21,861
News	1,804	2,402	4,206	4,206

Table 5: Number of positive, negative, total instances and number of instances containing at least one name for both downstream sentiment datasets.

6.2 Name Sentences

For the downstream analysis, we use *name sentences* that are sentences that contain person names from the name list. We use the list of base sentences shown in the left column of Table 6, which includes examples from both the Review and News datasets (e.g. *"Reviewed by Hugo"*), and an example that contains only a person's name. The tag [name] is replaced with each name from the name list, resulting in a total of 341,100 name sentences for the downstream analysis.

[4]We selected names of political persons from US politics (republicans and democrats).

	Reviews			*News*		
Name sentence	**SGNS**	**DebiasEmb**	**Mitigation effect**	**SGNS**	**DebiasEmb**	**Mitigation effect**
[name] applies for asylum	512	0	512	1038	485	553
[name] is head of the state	440	125	315	1539	811	728
[name] is an actress	331	56	275	1475	550	925
[name] is an actor	331	56	275	1279	696	583
[name] runs for president	418	179	239	850	502	348
[name] runs for governor	418	179	239	2438	1028	1410
[name]	389	201	188	814	221	593
[name] is played by [name]	293	119	174	1455	659	796
Reviewed by [name]	368	196	172	147	58	89
[name] is president	184	25	159	1313	377	936
[name] is governor	184	25	159	1525	901	624
[name] is ceo	184	25	159	1386	66	1320
The movie features [name]	235	77	158	1491	892	599
[name] lives in the us	165	7	158	1605	893	712
[name] applies for a job	155	0	155	1569	835	734
The soundtrack is composed by [name]	153	25	128	1642	834	808
The name of the main character is [name]	96	10	86	1303	526	777
[name] is an us citizen	59	4	55	1338	547	791
[name] is a movie character	40	2	38	1517	784	733
[name] plays a role	51	46	5	699	493	206
Mean	250.3	67.85	182.45	1321.15	607.9	713.25

Table 6: Number of name sentences in the minority class for SGNS and DebiasEmb and mitigation effect for all base sentences using the Reviews and the News datasets for model training.

6.3 Text-level Classifier

The text-level classifier takes the concatenated embedding representations of all words in a textual snippet and outputs a class probability score for each sentiment class. We use a Neural Network with three hidden layers, dropout (dropout rate = 0.5) and learning rate lr = 0.001, and train for 50 epochs.

As training data we use the **Reviews** and **News** datasets. However, before training, we applied NER to extract person names and replaced all names from our name list with an "unknown" token, making sure that the sentiment of names is not directly influenced by the training data. E.g., if a name would appear more often in negative instances than in positive instances in the training data, the classifier would be likely to associate it with negative sentiment, independently of the bias in the word embeddings. Therefore, filtering is necessary to make sure we measure biases in the *input embeddings* and not in the *training data*.

6.4 Downstream Bias Measures

An unbiased classifier towards names should classify a sentence independently of the person names it contains. For example, both sentences *"**Obama** applies for asylum"* and *"**Trump** applies for asylum"* should be placed into the same class (e.g. positive). That is, the classifier with a probability greater than 0.5 will place them in the same class, thus, showing no bias. However, if the change of name from "Obama" to "Trump" results in the change of the class probability, then the classifier is biased.

Hence, a bias free classifier would label all the name sentences with the same class. Name sentences that are *split* across sentiment classes represent a classification behavior that is biased. Correspondingly, we measure the bias of a downstream classifier as the amount of name sentences that are labelled with the minority class. In more details, if 80% of the name sentences are labelled as positive, we will consider the classifier to be biased towards the remaining 20% of name sentences that are labelled as negative in the minority class.

7 DOWNSTREAM ANALYSIS RESULTS

Table 7 shows the accuracy of the sentiment classifier using SGNS and DebiasEmb embeddings, and trained on the Reviews and News datasets. Similar to the word-level classifier, we note a small difference in terms of accuracy for DebiasEmb. The lower accuracy for News is explained by the smaller size of the dataset.

	Accuracy	
	Reviews	News
SGNS	0.84	0.77
DebiasEmb	0.83	0.73

Table 7: Text-classifier accuracy for SGNS and DebiasEmb on both datasets.

7.1 Highest and Lowest Ranked Names

Biased classifiers tend to give different sentiment labels and different class probability scores to sentences depending on the present names. Table 8 shows the highest and lowest probability scores that were given to names used in sentences by the SGNS model using embeddings trained on the *Random* news embeddings and

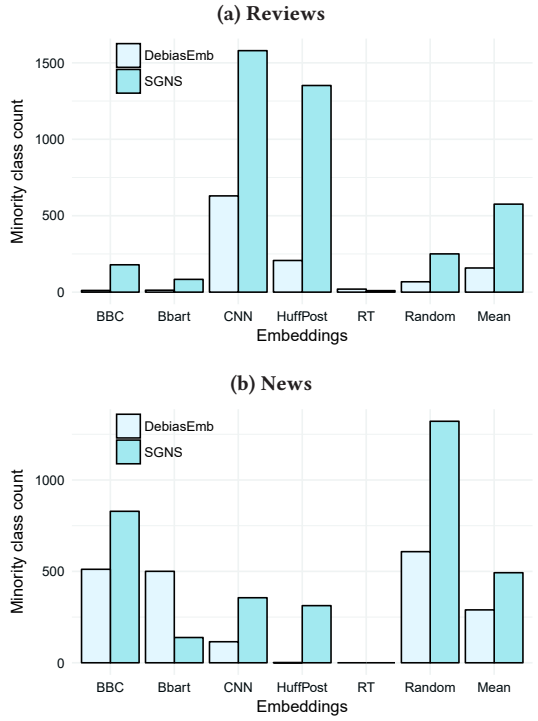

Figure 2: Average number of *name sentences* in the minority class for all embeddings trained with SGNS and DebiasEmb using the Reviews and the News datasets for model training. Ideally all variations of a name sentence are placed into one class, independently of the names used in the sentence, resulting in a low count for the minority class. DebiasEmb manages to reduce the mean count of the minority class compared to SGNS.

reviews for the sentiment classifier. Name rankings were consistent across different base sentences. The ranking shows that a sentence including the name *Kam* is much more likely to be placed in the positive sentiment class compared to the same sentence containing the name *Callahan*.

Name	High Pos. class prob.	Name	Low Pos. class prob.
Kam	0.644	...	...
Holder	0.599	Gere	0.346
Weisman	0.596	Silk	0.342
Portillo	0.596	Nino	0.333
Fax	0.591	Valentine	0.304
...	...	Callahan	0.299

Table 8: Highest and lowest class probabilities for the positive class averaged across all base sentences using the Random news embeddings, trained with SGNS on the Reviews dataset. A sentence including the name *Kam* is much more likely to be placed in the positive class, compared to the name *Callahan*.

7.2 Class Label Distribution

Figure 2 shows the average number of name sentences that are placed in the minority class when leveraging the embeddings trained on the different news datasets using SGNS and DebiasEmb for the (a) Reviews dataset, and (b) the News dataset. In case of the Reviews dataset, DebiasEmb reduces the number of minority class instances compared to SGNS for all news embeddings except for RussiaToday where the value is already very low for SGNS (9.85) and only slightly increased (19.80) for DebiasEmb. On average, DebiasEmb reduces the number of minority class instances from 575.68 to 158.06, resulting in an average mitigation effect of 417.62, meaning that an average of 417.62 names have changed the final label from the minority to the majority class.

When using the News dataset for training the sentiment classifier, we observe similar behavior with a mean mitigation effect of 203.41 from 492.73 to 289.32. For the embeddings trained on the Breitbart news dataset, we observe a negative effect. This is the only situation for News where we observed that DebiasEmb increases the minority class count instead of lowering it. For RussiaToday, we find the special case of having all instances already placed into the majority class when using the standard SGNS embeddings. DebiasEmb correctly keeps the class distribution and does not negatively impact the resulting labels.

Table 6 depicts the number of name sentences placed in the minority class for each base sentence using Random embeddings trained with SGNS and DebiasEmb. We observe a positive mitigation effect for all sentences for both the Reviews (average mitigation effect: 182.45) and the News dataset (average mitigation effect: 713.25). There is a stronger mitigation effect for some base sentences compared to others, though the domain of the sentence does not seem to have an impact as there are higher and lower mitigation values for both movie review and news sentences. For the Reviews dataset two of the base sentences are completely placed into one class, independently of the inserted name.

These results show that DebiasEmb increases the homogeneity of the final downstream class labels for similar sentences containing different names. This effect is independent of the domain that the downstream classifier is trained on. The diversity of the outcomes when using different datasets for training word embeddings shows that the choice of training data impacts not only the bias of word embeddings but also the potential of the debiasing approach.

8 DISCUSSION AND CONCLUSION

Bias in word embeddings stems mainly from the training corpora. In the case of textual corpora, depending on how such a corpus is created, it may contain various types of bias, following the theory that language reflects the attributes and norms of a societal group [10]. News in particular are mediators of *ideas, beliefs, ideology* [7]. Thus, for events, political actors, or other event actors (e.g. individuals of a specific community group), a common scheme in news discourse is language that is loaded with subjective words (i.e. positive or negative words) depending on the take a news source may have on a particular event.

Such insights are validated by several studies [3, 6], and similarly in this work too, we have seen that such bias stemming from the corpus, in our case sentiment bias associated with names, is present in word embeddings (SGNS), trained without taking any precaution

in terms of such biases. Furthermore, approaches that rely on predefined lexicons containing gender specific words against which target words are debiased, do not work due to the fact that such lexicons are incomplete and do not capture their proxies [6].

Considering such limitations, we proposed DebiasEmb, an approach for training and debiasing word embeddings based on the established method of skip-gram with negative sampling. Using an oracle sentiment classification model trained on words with explicit positive/negative sentiment, we address the problem of word proxies to consider the weights associated with the embedding dimensions and their values, rather than focusing solely on specific words. This allows us to take into account other word proxies in an automated manner. Apart from learning an accurate representation, DebiasEmb additionally aims at keeping each name embedding from a set of target names *indistinguishable* from both the positive and negative words and their proxies based on the oracle's classifier weights.

The evaluation results on word embeddings trained on varying news sources show that the sentiment bias associated with names is reduced significantly, while at the same time retaining high quality embeddings as measured by standard benchmarking results. Furthermore, on downstream tasks such as determining the sentiment of a text snippet, we showed that depending on the names present, models trained on various embeddings produce highly variable results. By using DebiasEmb the absolute majority of names is treated equally, resulting in a nearly 70% reduction of names that are discriminated by being categorized with a different label compared to the majority of the names.

Acknowledgments. This work is funded by the DESIR (grant no. 731081) and SimpleML (grant no. 01IS18054).

REFERENCES

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining.. In *Lrec*, Vol. 10. 2200–2204.

[2] Alexandra Balahur, Ralf Steinberger, Mijail A. Kabadjov, Vanni Zavarella, Erik Van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. 2013. Sentiment Analysis in the News. *CoRR* abs/1309.6202 (2013). arXiv:1309.6202 http://arxiv.org/abs/1309.6202

[3] 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*. 4349–4357.

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

[5] Mark Díaz, Isaac Johnson, Amanda Lazar, Anne Marie Piper, and Darren Gergle. 2018. Addressing age-related bias in sentiment analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 412.

[6] Yanai Elazar and Yoav Goldberg. 2018. Adversarial Removal of Demographic Attributes from Text Data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 11–21. https://aclanthology.info/papers/D18-1002/d18-1002

[7] Roger Fowler. 2013. *Language in the News: Discourse and Ideology in the Press*. Routledge.

[8] Maria Giatsoglou, Manolis G Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, and Konstantinos Ch Chatzisavvas. 2017. Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications* 69 (2017), 214–224.

[9] Hila Gonen and Yoav Goldberg. 2019. Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them. *arXiv preprint arXiv:1903.03862* (2019).

[10] Michael AK Halliday. 1970. Language structure and language function. *New horizons in linguistics* 1 (1970), 140–165.

[11] Zellig S Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.

[12] Christoph Hube and Besnik Fetahu. 2018. Detecting biased statements in wikipedia. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 1779–1786.

[13] Christoph Hube and Besnik Fetahu. 2019. Neural Based Statement Classification for Biased Language. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, New York, NY, USA, 195–203. https://doi.org/10.1145/3289600.3291018

[14] Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1113–1122.

[15] Stanisław Jastrzebski, Damian Leśniak, and Wojciech Marian Czarnecki. 2017. How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv:1702.02170* (2017).

[16] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open source scientific tools for Python. (2001–). http://www.scipy.org/ [Online; accessed <today>].

[17] Svetlana Kiritchenko and Saif M. Mohammad. 2018. Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems. *CoRR* abs/1805.04508 (2018). arXiv:1805.04508 http://arxiv.org/abs/1805.04508

[18] Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*. Springer, 415–463.

[19] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 142–150. http://www.aclweb.org/anthology/P11-1015

[20] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[21] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On Measuring Social Biases in Sentence Encoders. *CoRR* abs/1903.10561 (2019). arXiv:1903.10561 http://arxiv.org/abs/1903.10561

[22] 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*. 3111–3119.

[23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[25] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[26] Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1650–1659.

[27] Scharolta Katharina Sienčnik. 2015. Adapting word2vec to named entity recognition. In *Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania*. Linköping University Electronic Press, 239–243.

[28] Nathaniel Swinger, Maria De-Arteaga, IV Heffernan, Neil Thomas, Mark DM Leiserson, and Adam Tauman Kalai. 2018. What are the biases in my word embedding? *arXiv preprint arXiv:1812.08769* (2018).

[29] Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2015. Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network. *CoRR* abs/1510.06168 (2015). arXiv:1510.06168 http://arxiv.org/abs/1510.06168

[30] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender Bias in Contextualized Word Embeddings. (2019). arXiv:cs.CL/1904.03310

[31] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496* (2018).