DEBIE: A Platform for Implicit and Explicit Debiasing of Word Embedding Spaces

Niklas Friedrich, Anne Lauscher, Simone Paolo Ponzetto and Goran Glavaš
Data and Web Science Group
School of Business Informatics and Mathematics
University of Mannheim
nfriedri@mail.uni-mannheim.de
{anne,simone,goran}@informatik.uni-mannheim.de

Abstract

Recent research efforts in NLP have demonstrated that distributional word vector spaces often encode stereotypical human biases, such as racism and sexism. With word representations ubiquitously used in NLP models and pipelines, this raises ethical issues and jeopardizes the fairness of language technologies. While there exists a large body of work on bias measures and debiasing methods, to date, there is no platform that would unify these research efforts and make bias measuring and debiasing of representation spaces widely accessible. In this work, we present DEBIE, the first integrated platform for (1) measuring and (2) mitigating bias in word embeddings. Given an (i) embedding space (users can choose between the predefined spaces or upload their own) and (ii) a bias specification (users can choose between existing bias specifications or create their own), DEBIE can (1) compute several measures of implicit and explicit bias and modify the embedding space by executing two (mutually composable) debiasing models. DEBIE's functionality can be accessed through four different interfaces: (a) a web application, (b) a desktop application, (c) a REST-ful API, and (d) as a command-line application.[1] DEBIE is available at: debie.informatik.uni-mannheim.de

1 Introduction

Ethical and fair natural language processing is an essential precondition for widespread societal adoption of language technologies. In recent years, however, distributional language representations built from large corpora have been shown to encode human-like biases, like racism and sexism (Bolukbasi et al., 2016; Zhao et al., 2019; Lauscher et al., 2020a; Nadeem et al., 2020, *inter alia*). At the word level, most embedding spaces, across a range of embedding models and languages (Lauscher and Glavaš, 2019), encode human biases that can be exemplified in biased analogies, such as the famous example of sexism: $\overrightarrow{\text{man}} - \overrightarrow{\text{programmer}} \approx \overrightarrow{\text{woman}} - \overrightarrow{\text{homemaker}}$ (Bolukbasi et al., 2016). While this is not surprising, given the distributional nature of word representation models (Harris, 1954) it is – depending on the sociotechnical context – an undesired artefact of distributional representation learning (Blodgett et al., 2020) which can, in turn, lead to unfair decisions in downstream applications. A number of different measures for quantifying biases in representation spaces have been proposed in recent years (Caliskan et al., 2017; Gonen and Goldberg, 2019; Dev and Phillips, 2019; Garg et al., 2018; Lauscher et al., 2020a) and even more models for removing or attenuating such biases have been developed (Zhao et al., 2019; Bordia and Bowman, 2019; Dinan et al., 2020; Webster et al., 2020; Qian et al., 2019, *inter alia*). What is still missing, however, is the ability to seamlessly apply different bias measures and debiasing models on arbitrary embedding spaces and for custom (i.e., user-specified) bias specifications.

In this work, we address this gap by introducing DEBIE, the first integrated platform offering bias measurement and mitigation for arbitrary static embedding spaces and bias specifications. The DEBIE platform is grounded in the general framework for *implicit* and *explicit* debiasing of word embedding spaces (Lauscher et al., 2020a). Within this framework, an implicit bias consists of measurable discrepancies between two target term sets, which can, for instance, describe a dominant and a minoritized social group (D'Ignazio and Klein, 2020). In contrast, an explicit bias is a bias between such target term sets towards certain attribute terms groups. Our platform allows for both implicit and

[1] Videos demonstrating the usage of the DEBIE application and command-line tool are available at https://tinyurl.com/y2ymujus

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 91–98
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

explicit bias specifications, incorporating a range of different measures for quantifying embedding space bias (Caliskan et al., 2017; Gonen and Goldberg, 2019; Dev and Phillips, 2019) and a pair of mutually composable methods for bias mitigation. DEBIE's functionality for measuring and mitigating biases in distributional word vector spaces is accessible via four different interfaces: as a web application, desktop application, via a RESTful application programming interface (API), and as a command-line tool. We believe that *DebIE* will, by offering to test arbitrary embedding spaces for custom user-defined biases, stimulate a wider exploration of the presence of a broader set of human biases in distributional representation spaces.

2 Related Work

First, we describe related research on bias evaluation and debiasing and then turn our attention to existing bias mitigation platforms.

Bias Measures and Mitigation Methods. There is an extensive body of research on bias detection and bias mitigation in natural language processing. Due to space limitations, here we only provide a brief overview and refer the reader to a recent survey of the field for more information (Blodgett et al., 2020). Bolukbasi et al. (2016) were the first to show stereotypical bias to exist in word embedding models and proposed *hard debiasing*, the first word embedding bias mitigation algorithm. Subsequently, Caliskan et al. (2017) introduced the well-known Word Embedding Association Test (WEAT), inspired by the Implicit Association Test (Nosek et al., 2002), which measures biased associations in human subjects in terms of response times when exposed to sets of stimuli. WEAT, in turn, reflects the strength of associations in terms of semantic similarity between word vectors. McCurdy and Serbetci (2017) study gender bias with WEAT in three other languages (Dutch, German, and Spanish). Extending upon this, Lauscher and Glavaš (2019) translated the WEAT tests to 6 more languages (German, Spanish, Italian, Russian, Croatian, Turkish), allowing for multilingual and cross-lingual analysis of biases captured by the specifications of the original WEAT. They later extended the set of supported languages with Arabic (Lauscher et al., 2020b).

Dev and Phillips (2019) proposed a linear projection model for debiasing along with two bias evaluation measures: the Embedding Coherence Test (ECT) and the Embedding Quality Test (EQT) and propose methods for removing the (explicit) bias based on computing the direction vector of the bias. While their method successfully removes the *explicit* bias, i.e., bias between sets of *target* terms (e.g., male terms like *man*, *father*, and *boy* vs. female terms like *woman*, *mother*, and *girl*) *with respect to* sets of *attribute* terms (e.g., profession terms, such as *scientist* or *artist*), Gonen and Goldberg (2019) show that *implicit* bias between the sets of target terms remains even after (explicit debiasing) and that the terms from one target set are still clearly discernible from the terms of the other set in the embedding space. Based on this finding, Lauscher et al. (2020a) systematized the preceding work and proposed a general framework for bias measurement and debiasing, encompassing a range of existing and newly proposed measures and mitigation methods, which operate either on *explicit* or *implicit* bias specifications. Their framework arguably allows for a more holistic assessment of bias in word vector spaces and ensures interoperability between bias mitigation models and bias specifications. Our DEBIE platform makes this holistic framework for measuring and mitigating biases widely accessible and applicable (1) for arbitrary user-defined bias specification to (2) arbitrary pretrained word embedding spaces.

Bias mitigation platforms. The landscape of the *off-the-shelf* solutions for measuring and mitigating bias for machine learning applications is extremely scarce. To the best of our knowledge, the only such tool is *AI Fairness 360* (Bellamy et al., 2018), an extensible open-source toolkit which offers a set of algorithms for detecting and mitigating unwanted bias in datasets and machine learning models. It addresses bias by integrating fairness algorithms along the machine learning pipeline, i.e., fair pre-processing, fair in-processing, and fair post-processing. In contrast, DEBIE specifically targets biases in distributional word vector spaces (as an ubiquitous component of modern NLP pipelines) by integrating a series of word embedding bias tests and mitigation algorithms not covered by more general tools like AI Fairness 360.

3 DEBIE: System Description

We first explain the two types of bias specifications support by DEBIE (*implicit* and *explicit*), then proceed to describe the concrete bias specifications and debiasing algorithms bundles in the system. Finally, we provide details of DEBIE's architecture and interfaces through which the bias measuring and mitigation functionality can be accessed. All code is publicly available on GitHub.[2]

3.1 Implicit and Explicit Bias Specifications

DEBIE supports measuring of *implicit* or *explicit* biases for a given word embedding space and, respectively, implicit or explicit *debiasing* of the given space. Both implicit bias specifications B_I and explicit bias specifications B_E specify two sets of *target* terms, T_1 and T_2 that capture the dimension of the bias. For example, if measuring a *gender* bias, T_1 would contain male terms (e.g., *man*, *father*) and T_2 female terms (e.g., *woman*, *girl*, *grandma*).[3] While an implicit bias specification is fully specified with the two target lists, $B_I = (T_1, T_2)$, an explicit specification additionally requires two sets of attributes A_1 and A_2, $B_E = (T_1, T_2, A_1, A_2)$, capturing the groups of terms towards which the target groups are expected to exhibit significantly different level of association. For example, for a gender bias, one would expect male terms to be more strongly associated with career terms (e.g., A_1 could contain terms like *programmer*), whereas female terms could be closer to family-related terms (e.g., A_2 could contain terms like *homemaker*). The input for DEBIE consists of an embedding space $\mathbf{X} \in \mathbb{R}^d$ and a bias specification, (implicit or explicit). Explicit debiasing methods (i.e., methods that operate on explicit bias specifications) cannot be executed when the provided bias specification is implicit (B_I).[4]

3.2 Bias Measures

DEBIE provides three measures that capture explicit bias (i.e., apply only if an explicit bias specification is provided), and two tests that measure implicit bias. Because debiasing methods (see §3.3) make perturbations to the embedding space, we additionally couple the bias tests with measures of semantic quality of the distributional space.

Word Embedding Association Test (WEAT). Given an explicit bias test specification $B_E = (T_1, T_2, A_1, A_2)$, WEAT (Caliskan et al., 2017) computes the effect size quantifying the amount of bias as follows:

$$s(T_1, T_2, A_1, A_2) = \sum_{t_1 \in T_1} s(t_1, A_1, A_2) - \sum_{t_2 \in T_2} s(t_2, A_1, A_2),$$

with associative difference of term t given as:

$$s(t, A_1, A_2) = \frac{1}{|A_1|} \sum_{a_1 \in A_1} \cos(\mathbf{t}, \mathbf{a_1}) - \frac{1}{|A_2|} \sum_{a_2 \in A_2} \cos(\mathbf{t}, \mathbf{a_2}),$$

with $\mathbf{t}$ as the word embedding of the target term t and cos as the cosine of the angle between the two vectors. To estimate the significance of the effect size, we follow Caliskan et al. (2017) and compute the non-parametric permutation test in which the $s(T_1, T_2, A_1, A_2)$ is compared to $s(X_1, X_2, A_1, A_2)$, where (X_1, X_2) denotes a random, equally-sized split of terms from $T_1 \cup T_2$.

Embedding Coherence Test (ECT). Given an explicit bias specification with a single attribute set $B_E = (T_1, T_2, A)$ with $A = A_1 \cup A_2$, ECT (Dev and Phillips, 2019) quantifies the presence of the bias as the (lack of) correlation of the distances of the mean vectors of the target term sets T_1 and $T2$ with the attribute terms in A. The lower the correlation, the higher the bias. To this end, we compute the mean vectors $\mathbf{t_1}$ and $\mathbf{t_2}$ as averages of the vector representations of the terms in T_1 and T_2. Next, we compute two vectors containing the cosine similarities of each of the terms in A with $\mathbf{t_1}$, as well as with $\mathbf{t_2}$, respectively. The final score is Spearman's rank correlation coefficient of the obtained vectors of cosine similarity scores.

Bias Analogy Test (BAT). BAT (Lauscher et al., 2020a) assesses the amount of biased analogies that can be retrieved from an embedding space based on the explicit bias specification $B_E = (T_1, T_2, A_1, A_2)$. We first create all possible biased analogies from B_E: $\mathbf{t}_1 - \mathbf{t}_2 \approx \mathbf{a}_1 - \mathbf{a}_2$ for $(t_1, t_2, a_1, a_2) \in T_1 \times T_2 \times A_1 \times A_2$. Next, from each of these analogies, two query vectors are computed: $\mathbf{q}_1 = \mathbf{t}_1 - \mathbf{t}_2 + \mathbf{a}_2$ and $\mathbf{q}_2 = \mathbf{a}_1 - \mathbf{t}_1 + \mathbf{t}_2$ for each 4-tuple (t_1, t_2, a_1, a_2). We then rank all attribute vectors in $\mathbf{X}$ according to the Euclidean distance to the query vector. We report the percentage of cases in which: (1) a_1 is ranked higher than a term $a'_2 \in A_2 \setminus \{a_2\}$ for $\mathbf{q}_1$ and (2) a_2 is ranked higher than a term $a'_1 \in A_1 \setminus \{a_1\}$ for $\mathbf{q}_2$.

[2] https://github.com/umanlp/debie-frontend
https://github.com/umanlp/debie-backend

[3] The bias measures implemented in DEBIE do not require the terms between the target lists to be paired. Accordingly, the two lists also do not need to be of the same length.

[4] Conversely, implicit debiasing methods, i.e., ones that require only T_1 and T_2, can be applied if an explicit specification is provided. In that case, we simply convert B_E to B_I by discarding the provided attribute sets.

Implicit Bias Tests (IBT). As proposed by Gonen and Goldberg (2019), the amount of implicit bias corresponds to the accuracy with which two target term sets can be separated. We report the score of two methods: (1) clustering accuracy with K-Means++ (Arthur and Vassilvitskii, 2007), and (2) classification accuracy based on Support Vector Machines with Gaussian kernel. We carry out the latter via leave-one-out cross-validation (i.e., we train on all words from both target lists, leaving one term for prediction).

Semantic Quality Tests (SQ). The debiasing models (3.3) modify the embedding space. While they reduce the bias, they may reduce the general semantic quality of the embedding space, which could be detrimental for model performance in downstream applications. This is why we couple the bias tests with measures of semantic word similarity on two established word-similarity datasets: SimLex-999 (Hill et al., 2015) or WordSim-353 (Finkelstein et al., 2001). We compute the Spearman correlation between the human similarity scores assigned to word pairs and corresponding cosines computed from the embedding space.

3.3 Debiasing Methods

DEBIE encompasses implementations of two debiasing models from (Lauscher et al., 2020a), for which an implicit bias specification suffices:[5]

General Bias Direction Debiasing (GBDD). As an extension of the linear projection model of Dev and Phillips (2019), GBDD relies on identifying the bias direction in the distributional space. Let (t_1^i, t_2^j) be word pairs with $t_1^i \in T_1$, $t_2^j \in T_2$, respectively. First, we obtain partial bias direction vectors $\mathbf{b_{ij}}$ by computing the difference between the respective vectors for each pair $\boldsymbol{b_{ij}} = \boldsymbol{t_1^i} - \boldsymbol{t_2^j}$. We then stack all partial direction vector, obtaining the bias matrix $\boldsymbol{B}$. The global bias direction vector $\boldsymbol{b}$ then corresponds to the top singular value of $\boldsymbol{B}$, i.e., the first row of matrix $\boldsymbol{V}$, with $\boldsymbol{U\Sigma V}^\top$ as the singular value decomposition of $\boldsymbol{B}$. We then obtain the debiased version of the space $\boldsymbol{X}$ as:

$$\text{GBDD}(\mathbf{X}) = \mathbf{X} - \langle \mathbf{X}, \mathbf{b} \rangle \mathbf{b},$$

with $\langle \boldsymbol{X}, \boldsymbol{b} \rangle$ denoting dot products between rows of $\boldsymbol{X}$ and $\boldsymbol{b}$. As such, the closer the word embedding is to the bias direction, the more it gets corrected.

Bias Alignment Model (BAM). Inspired by previous work on projection-based cross-lingual word embedding spaces (Smith et al., 2017; Glavaš et al., 2019), BAM focuses on *implicit* debiasing by treating the target term sets T_1, and T_2 of an implicit bias specification B_I as "translations" of each other and learning the linear projection of the embedding spaces w.r.t. itself (Lauscher et al., 2020a). First, we build all possible word pairs (t_1^i, t_2^j), $t_1^i \in T_1$, $t_2^j \in T_2$ and stack the respective word vectors of the left and right pairs to obtain matrices $\boldsymbol{X}_{T_1}$ and $\boldsymbol{X}_{T_2}$. We then learn the orthogonal mapping matrix $\boldsymbol{W_X} = \boldsymbol{UV}^\top$, with $\boldsymbol{U\Sigma V}^\top$ as the singular value decomposition of $\boldsymbol{X}_{T_2}\boldsymbol{X}_{T_1}^\top$. In the last step, the original space and its "translation" $\boldsymbol{X} = \boldsymbol{XW_X}$ (which is equally biased), are averaged to obtain the debiased embedding space:

$$\text{BAM}(\boldsymbol{X}) = \frac{1}{2}(\boldsymbol{X} + \boldsymbol{XW_X})\,.$$

Note that DEBIE can trivially compose the two debiasing models – the resulting space after applying GBDD (BAM) can be the input for BAM (GBDD).

3.4 Integrated Data

DEBIE is designed as a general tool, which allows user to upload their own embedding spaces and define their own bias specifications for testing and/or debiasing. Nonetheless, we include into the platform a set of commonly used bias specifications and word embedding spaces. Concretely, DEBIE includes the whole WEAT test collection (Caliskan et al., 2017), containing the explicit bias specifications summarized in Table 1. DEBIE also comes with three word embedding spaces, pretrained with different models: (1) fastText (Bojanowski et al., 2017),[6] (2) GloVe (Pennington et al., 2014),[7] and (3) CBOW (Mikolov et al., 2013).[8] All three spaces are 300-dimensional and their vocabularies are limited to 200K most frequent words.

3.5 System Architecture

DEBIE's architecture, illustrated in Figure 1, adheres to the principles of modern extensible web application design and consists of four components: (1) the backend, (2) the frontend, which together

[5] Note that any explicit bias specification is trivially reduced to an implicit one by discarding the attribute term sets.

[6] https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.en.vec

[7] http://nlp.stanford.edu/data/glove.6B.zip

[8] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

Test	Type	Target Set #1	Target Set #2	Attribute Set #1	Attribute Set #2
1	Universal	Flowers (e.g., *aster*, *tulip*)	Insects (e.g., *ant*, *flea*)	Pleasant (e.g., *health*, *love*)	Unpleasant (e.g., *abuse*)
2	Militant	Instruments (e.g., *cello*, *guitar*)	Weapons (e.g., *gun*, *sword*)	Pleasant	Unpleasant
3	Racist	Euro-American names (e.g., *Adam*)	Afro-American names (e.g., *Jamel*)	Pleasant (e.g., *caress*)	Unpleasant (e.g., *abuse*)
4	Racist	Euro-American names (e.g., *Brad*)	Afro-American names (e.g., *Hakim*)	Pleasant	Unpleasant
5	Racist	Euro-American names	Afro-American names	Pleasant (e.g., *joy*)	Unpleasant (e.g., *agony*)
6	Gender	Male names (e.g., *John*)	Female names (e.g., *Lisa*)	Career (e.g. *management*)	Family (e.g., *children*)
7	Gender	Math (e.g., *algebra*, *geometry*)	Arts (e.g., *poetry*, *dance*)	Male (e.g., *brother*, *son*)	Female (e.g., *woman*, *sister*)
8	Gender	Science (e.g., *experiment*)	Arts	Male	Female
9	Disease	Physical condition (e.g., *virus*)	Mental condition (e.g., *sad*)	Long-term (e.g., *always*)	Short-term (e.g., *occasional*)
10	Age	Older names (e.g., *Gertrude*)	Younger names (e.g., *Michelle*)	Pleasant	Unpleasant

Table 1: WEAT bias test specifications provided by DEBIE.

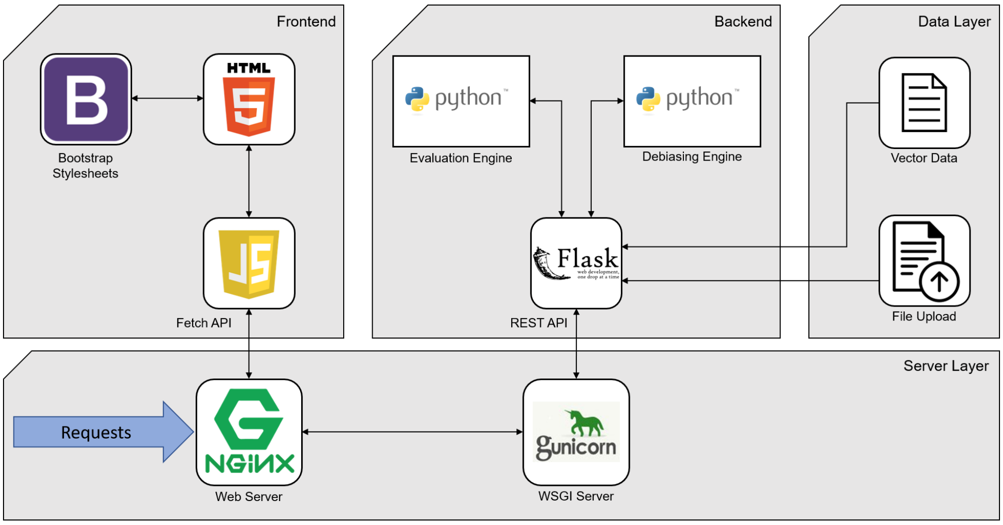


Figure 1: Software Architecture of the DEBIE platform.

represent the core of the application, (3) the data layer, and (4) the server layer facing the web.

Backend. DEBIE's backend consists of two main modules: (1) the bias evaluation engine, which computes the bias test scores (see §3.2), and (2) the debiasing engine, which runs the word embedding debiasing models (see §3.3). The backend interacts with the data layer for retrieving data (bias specifications and vectors from embedding spaces) and its functionality is exposed via a RESTful API, which offers endpoints for programmatically (i) uploading and retrieving data as well as for (ii) running bias evaluation and (iii) debiasing.

There are dedicated controllers and handlers for each of this primary functionalities: vector retrieval, bias evaluation, and debiasing. These are responsible for computing results and delivering content to relevant web pages. The second group of controllers and handlers is responsible for retrieving data out of integrated and external embedding spaces and for parsing and generating JSON data. All bias measures and debiasing methods are implemented as separate modules so that the platform can be extended seamlessly with additional bias measures and debiasing models. A new bias measure or a new debiasing model can be integrated by simply adding the computation scripts (i.e., a function that implements the functionality) and adapting the responsible handler. The backend is purely implemented in Python.

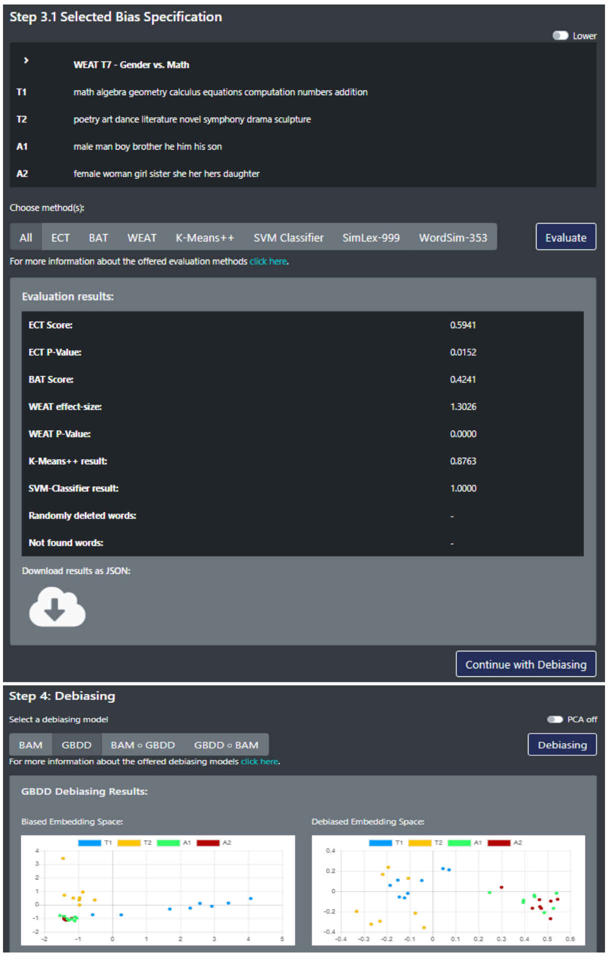
Frontend, Data Layer, and Server Layer. The frontend is written in HTML and plain JavaScript, and relies on the Bootstrap library.[9] The `fetch` functionality is used for sending requests to the RESTful API of the backend. For the visualization of embedding spaces (see bottom part of Figure 2), we rely on the the `chart.js` library.[10]

Embedding spaces are stored as two files: (1) the *.vocab* file is the serialized dictionary that maps words to indices of the embedding matrix; (2) the *.vectors* file is an embedding matrix (serialized 2D `numpy` array) rows of which are the actual word vectors. At the start of the web applicatiob, all bias specifications and intergated embedding spaces are fully loaded into the memory completely.

DEBIE is hosted on a Linux server, running De-

[9] https://getbootstrap.com/

[10] https://www.chartjs.org/



Step 3.1 Selected Bias Specification

	WEAT T7 - Gender vs. Math
T1	math algebra geometry calculus equations computation numbers addition
T2	poetry art dance literature novel symphony drama sculpture
A1	male man boy brother he him his son
A2	female woman girl sister she her hers daughter

Choose method(s):

For more information about the offered evaluation methods click here.

Evaluation results:

ECT Score:	0.5941
ECT P-Value:	0.0152
BAT Score:	0.4241
WEAT effect-size:	1.3026
WEAT P-Value:	0.0000
K-Means++ result:	0.8763
SVM-Classifier result:	1.0000
Randomly deleted words:	-
Not found words:	-

Download results as JSON:

Step 4: Debiasing

Select a debiasing model

For more information about the offered debiasing models click here.

GBDD Debiasing Results:

Biased Embedding Space:

Debiased Embedding Space:

Figure 2: DEBIE’S web UI.

bian 10 as the operating system. The python WSGI-server `gunicorn` is used to serve the RESTful API. We opt for `nginx` as the web server for hosting the frontend and redirecting the API-requests to the internal endpoints of the WSGI-server.

3.6 Accessibility: Interfaces

Users can interact with DEBIE through four different interfaces. The simplest way is by using the provided web interface. For programmatic access, we offer the RESTful-API accessible directly via HTTP requests. As a third option, a desktop version of the tool is available for download: this tools runs completely offline and, depending on the hardware, may perform faster. Finally, we offer a command-line interface intended for shell usage.

Web User Interface. DEBIE is primarily imagined as a web application with a full extendable web user interface (see Figure 2). The web-UI enables users to evaluate and debias with predefined or custom bias specifications. Designed as a *one-page application*, the web UI guides the user via five simple steps through the full process:

Step 1: Selection of the Embedding Space. In the first step, the user has to select with which embedding space to work. The users can select one of three integrated embedding spaces (§3.4) uploaded or their own pretrained vector space.

Step 2: Selection of the Bias Specification. The user next chooses a bias specification: they can select one of the integrated WEAT bias specifications or define a bias specification of their own.

Step 3: Selection and Computation of the Bias Tests. The user next selects bias measures/scores (see §3.2) to be applied on the selected embedding space given the selected bias specification. The bias (and semantic similarity) scores are displayed in a table (see the upper part of Figure 2) and can also be exported as in the JSON format.

Step 4: Selection and Execution of Debiasing Algorithms. The user can next choose to debias the selected embedding space (Step 1) based on the selected bias specification (Step 2). To this effect, the user can choose between GBDD, BAM, or one of their compositions (GBDD∘BAM or BAM∘GBDD). The debiased embeddings space can be downloaded. To visualize the differences between the original (biased) and debiased embedding space, we visualize the 2D PCA-compressions of the terms from the bias specification in both spaces (see bottom part of Figure 2).

Step 5: Computation of Bias Tests on the Debiased space. Finally, the user can evalute the effects of debiasing with the desired set of bias measures. This is like Step 3, only now we subject to testing the debiased instead of the original embedding space.

RESTful API. For programmatic access, we offer a RESTful API. The API can deliver vector representations of words, compute and fetch the bias evaluation scores, as well as debiased word embeddings based on a provided bias specification. The API endpoints are accessible online.[11] API documentation is available in the `swagger` format on the DEBIE website.[12]

Desktop Application. We offer an adapted offline-version of the web application providing the same functionality, runnable on Windows OS. The desktop app has been created with the python module `flaskwebgui`, using the source files of

[11] http://debie.informatik.uni-mannheim.de:8000/REST/

[12] http://debie.informatik.uni-mannheim.de:8000/swagger/

the web application. The desktop application is available both as a windows executable file (.exe) and as a python script.

Command-line Interface. Finally, we expose DEBIE's functionality through a command-line interface, intended for shell (e.g., bash) usage. We employ the Python framework click to parse the command line arguments.

4 Ethical Considerations

Given the high sensitivity of the issue of bias in text representations, we would like the reader to consider the following three aspects.

(i) Our platform allows for measuring and mitigating biases based on bias specifications, which need to be defined by the user. In actual deployment scenarios, those specifications need to be designed with extreme care and the concrete sociotechnical environment in mind. For instance, it would be wrong to assume that by using one of the predefined gender bias specifications provided with this platform, all stereotypical gender associations will be removed from the representation space. In contrast, for each individual application scenario, the user should make sure that the bias specification matches the bias evaluation and debiasing intent.

(ii) Though the user's main role is to choose appropriate bias specifications, we think it is important that the user has enough technical proficiency to understand potential issues of the provided measures and mitigation methods.

(iii) The gender bias specifications from previous work provided with this platform only consider bias between *male* and *female* term sets, i.e., they follow a binary notion of gender. However, it is important to keep in mind that gender is a spectrum. We fully acknowledge the importance of the inclusion of *all gender identities*, e.g., nonbinary, gender fluid, polygender, etc., in language technologies.

5 Conclusion

We have presented DEBIE, an integrated platform for measuring and attenuating implicit and explicit biases in distributional word vector spaces. Via four different interfaces, we enable fast and easy access to a variety of bias measures and debiasing methods, allowing users to experiment with arbitrary embedding spaces and bias specifications. We hope DEBIE facilitates an exploration of a wider set of human biases in language representations.

Acknowledgments

Anne Lauscher and Goran Glavaš are supported by the Eliteprogramm of the Baden-Württemberg Stiftung (AGREE grant). We would like to thank the anonymous reviewers for their helpful comments.

References

David Arthur and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of SODA*, pages 1027–1035.

Rachel Bellamy, Kuntal Dey, Michael Hind, Samuel Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Ramazon Kush, and Yunfeng Zhang. 2018. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in nlp. In *Proceedings of the 58th Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

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

Tolga Bolukbasi, Kai Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. pages 4356–4364.

Shikha Bordia and Samuel Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15.

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

Sunipa Dev and Jeff Phillips. 2019. Attenuating bias in word vectors. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 879–887. PMLR.

Catherine D'Ignazio and Lauren F Klein. 2020. The power chapter. In *Data Feminism*. The MIT Press.

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020. Queens are powerful too: Mitigating gender bias in dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8173–8188, Online. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):3635–3644.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of ACL*, pages 710–721.

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. In *Proceedings of NAACL-HLT*, pages 609–614.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Anne Lauscher and Goran Glavaš. 2019. Are We Consistently Biased? Multidimensional Analysis of Biases in Distributional Word Vectors. pages 85–91.

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

Anne Lauscher, Rafik Takieddin, Simone Paolo Ponzetto, and Goran Glavaš. 2020b. AraWEAT: Multidimensional analysis of biases in Arabic word embeddings. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 192–199, Barcelona, Spain (Online). Association for Computational Linguistics.

Katherine McCurdy and Oguz Serbetci. 2017. Grammatical gender associations outweigh topical gender bias in crosslinguistic word embeddings. In *Proceedings of WiNLP*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*, pages 3111–3119.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. 2019. Reducing gender bias in word-level language models with a gender-equalizing loss function. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 223–228.

Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 629–634.