Fair Embedding Engine: A Library for Analyzing and Mitigating Gender Bias in Word Embeddings

Vaibhav Kumar* **Tenzin Singhay Bhotia*** **Vaibhav Kumar***
Delhi Technological University
Delhi, India
{kumar.vaibhav1o1, tenzinbhotia0, vaibhavk992}@gmail.com

Abstract

Non-contextual word embedding models have been shown to inherit human-like stereotypical biases of gender, race and religion from the training corpora. To counter this issue, a large body of research has emerged which aims to mitigate these biases while keeping the syntactic and semantic utility of embeddings intact. This paper describes Fair Embedding Engine (FEE), a library for analysing and mitigating gender bias in word embeddings. FEE combines various state of the art techniques for quantifying, visualising and mitigating gender bias in word embeddings under a standard abstraction. FEE will aid practitioners in fast track analysis of existing debiasing methods on their embedding models. Further, it will allow rapid prototyping of new methods by evaluating their performance on a suite of standard metrics.

1 Introduction

Non-contextual word embedding models such as Word2Vec (Mikolov et al., 2013b,a), GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017) have been established as the cornerstone of modern natural language processing (NLP) techniques. The ease of usage followed by performance improvements (Turian et al., 2010) have made word embeddings pervasive across various NLP tasks. However, as with most things, the gains come at a cost, word embeddings also pose the risk of introducing unwanted stereotypical biases in the downstream tasks. Bolukbasi et al. (2016a) showed that a Word2Vec model trained on the Google news corpus, when evaluated for the analogy *man:computer programmer :: woman:?* results to the answer *homemaker*, reflecting the stereotypical biases towards woman. Further, Zhao et al. (2018a) showed that models operating on biased word embeddings can leverage stereotypical cues in downstream tasks like co-reference resolution as heuristics to make thier final predictions.

Addressing the issues of unwanted biases in learned word representations, recent years have seen a surge in the development of word embedding debiasing procedures. The fundamental aim of a debiasing procedure is to mitigate stereotypical biases while introducing minimal semantic offset, hence maintaining the usability of embeddings. Based upon the mode of operation, the debiasing methods can be classified into two categories: First, post-processing methods, which operate upon pre-trained word vectors (Bolukbasi et al., 2016a; Kaneko and Bollegala, 2019; Yang and Feng, 2020). Second, learning based methods, which involve re-training the word embedding models by either making changes to the training data or to the training objective. (Zhao et al., 2018b; Lu et al., 2018; Bordia and Bowman, 2019). Along with the development of debiasing procedures, numerous metrics to evaluate the efficacy of each debiasing procedure have also been proposed (Zhao et al., 2018b; Bolukbasi et al., 2016a; Kumar et al., 2020). Although the domain has largely benefited from the contributions of different researchers, the domain still lacks open source software projects that unify such diverse but fundamentally similar methods in an organized and standard manner. Therefore, the domain has a high barrier for newcomers to overcome, and the domain experts may still need to put in extra effort of building and maintaining their own codebases.

To solve this problem, we introduce Fair Embedding Engine (FEE), a library which combines state of the art techniques for debiasing, quantifying and visualizing gender bias in non-contextual word embeddings for the English language. The goal of FEE is to serve as a unified framework towards the

*Authors have contributed equally.

Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), pages 26–31
Virtual Conference, November 19, 2020. ©2020 Association for Computational Linguistics

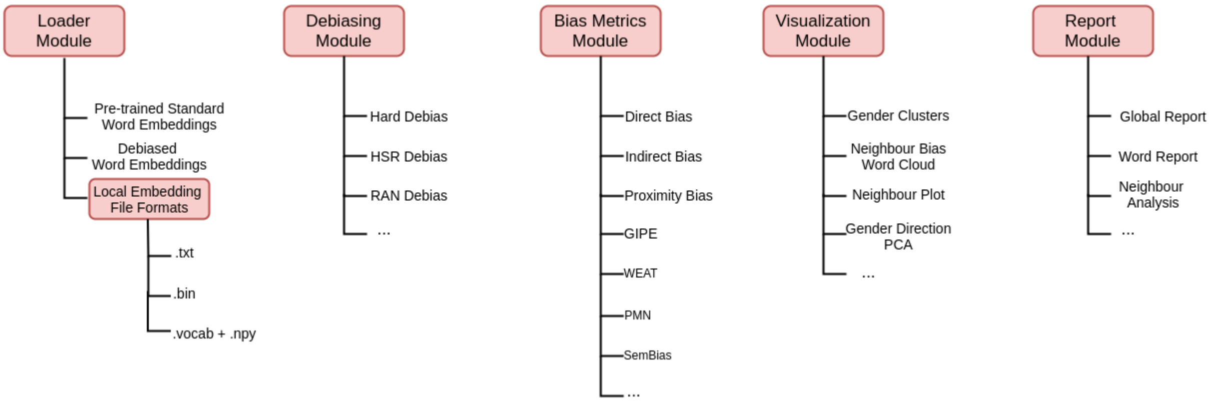


Figure 1: An inventory of implemented methods under four major modules that constitute FEE. Out of the box, FEE provides a subset of the prominent methods in each the modules. Further, each module can be easily extended to incorporate latest methods.

analysis of biases in word embeddings and the efficient development of better debiasing and bias evaluation methods. Conforming to the common style of implementation in some existing research works (Bolukbasi et al., 2016b; Zhao et al., 2018b), we use Numpy (Oliphant, 2006; Van Der Walt et al., 2011) arrays to store word vectors while keeping an index mapping to the strings of corresponding words. Further, we use the PyTorch (Paszke et al., 2017) Autograd engine for gradient based optimization and Matplotlib (Hunter, 2007) for generating plots. FEE is made available at: https://github.com/FEE-Fair-Embedding-Engine/FEE.

2 Related Work

Since the study of stereotypical biases in NLP has received attention only in the recent years, the domain has not been a part of open source efforts that attempt to integrate diverse sets of independent methods. The only relevant open source software (OSS) that we came across during our investigation was Word Embedding Fairness Evaluation (WEFE) framework (Badilla et al., 2020). For a given collection of pre-trained word embedding and a set of fairness criteria, WEFE ranks the embeddings based on their performance on an encapsulation of the fairness metrics. In order to achieve this ranking over an otherwise disparate set of fairness metrics (WEAT (Caliskan et al., 2017a), RND (Garg et al., 2018), and RNSB (Sweeney and Najafian, 2019)) WEFE introduces an abstraction which generalizes over the metrics using a set of *target* (the intended social class for which fairness is to be evaluated) and *attribute* words (the traits over which bias might exist for the selected target words). Further, (Badilla et al., 2020) conclude that while existing fairness metrics show a strong correlation when used for evaluating gender bias, only a weak correlation results when evaluating biases like religion and race.

Therefore, the focus of WEFE is limited to the evaluation of pre-trained word vectors on a suite of fairness metrics, lacking any support for debiasing methods. Further, only those evaluation metrics can be used which comply with the abstraction. FEE, on the other hand, provides holistic functionality by equipping a suite of evaluation and debiasing methods, along with a flexible design to assist researchers in developing new solutions.

FEE currently offers three debiasing methods as a part of its debiasing module: HardDebias (Bolukbasi et al., 2016b), HSRDebias (Yang and Feng, 2020), and RANDebias (Kumar et al., 2020). The bias metrics module consist of the following: SemBias (Zhao et al., 2018b), direct and indirect bias (Bolukbasi et al., 2016a), Gender-basied Illicit Proximity Estimate (GIPE) and Proximity bias (Kumar et al., 2020), Percent Male Neighbours (PMN) (Gonen and Goldberg, 2019) and Word Embedding Association Test (WEAT) (Caliskan et al., 2017b).

3 Fair Embedding Engine

The core functionality of FEE is governed by five modules, namely *Loader*, *Debias*, *Bias Metrics*, *Visualization*, and *Report*. Figure 1 illustrates the components for each module of FEE. In the following subsections, we delineate upon the implementation of each of the modules along with the motivation for their development.

3.1 Loader Module

Motivation: The foremost step in the analysis of word embeddings is to load them into the random access memory. However, different formats of local embedding files, and heterogeneous formats of pre-trained embedding sources may entail disparate forms of access, making the loading process non-trivial. The loader module abstracts this preprocessing step and provides a standardized object based access of word embeddings to its users.

Working: The workhorse of the loader module is its Word Embedding class, `WE`. Any version of a word embedding model can be considered as a unique instance of the `WE` class. It consists of a user accessible `loader()` method that either takes in an embedding name representing a pre-trained word embedding, or a local embedding file path as input and returns an initialized `WE` object. We integrate the well established Gensim (Řehůřek and Sojka, 2010) API in our loader module for providing access to several pre-trained embeddings. However, since FEE focuses on the bias domain, it also provides the functionality to either store the debiased counterparts of Gensim-loaded embeddings, or load an externally downloaded debiased embedding file. For flexibility, the loader module supports three prominent file formats i.e. `.txt`, `.bin`, and `.vocab` (words) + `.npy` (vectors). Once, a `WE` object is initialized with an embedding version via the `loader()` method, a user can obtain the vector representation for a word by calling its vector method, `v()` with that word as its argument. All the subsequent modules of FEE operate on the `WE` object for achieving their objectives.

3.2 Debiasing Module

Motivation: The domain of bias in word representations considers effective debiasing methods as one of their ultimate objectives. Much effort has been made in the recent years to develop good debiasing methods. However, most works flaunt the efficacy of their debiasing procedures by applying them to a limited number of pre-trained embeddings. We hope that future works try to experiment their new methods or the existing ones on different embeddings. However, such a task involves refactoring and modification of individually tailored prior works. The debiasing module of FEE re-implements these diverse algorithms and provides a standardized access to users while facilitating reproducible research.

Working: The debiasing module of FEE currently provides access to some of the proposed post-processing debiasing procedures in the past, as shown in Figure 1. Each debiasing method is represented by a unique class in the module. For instance, the Hard Debias methods proposed by Bolukbasi et al. (2016a) is assigned a class named, `HardDebias`. Since all debiasing methods are fundamentally applied to a word embedding, the class of each debiasing method is initialised by a `WE` object. Each debiasing class has a common method called `run()` that takes in a list of words as argument and runs the entire debiasing procedure on it. As the debiasing procedure operates upon the `WE` object, the engineering effort in dealing with different embedding formats is mitigated.

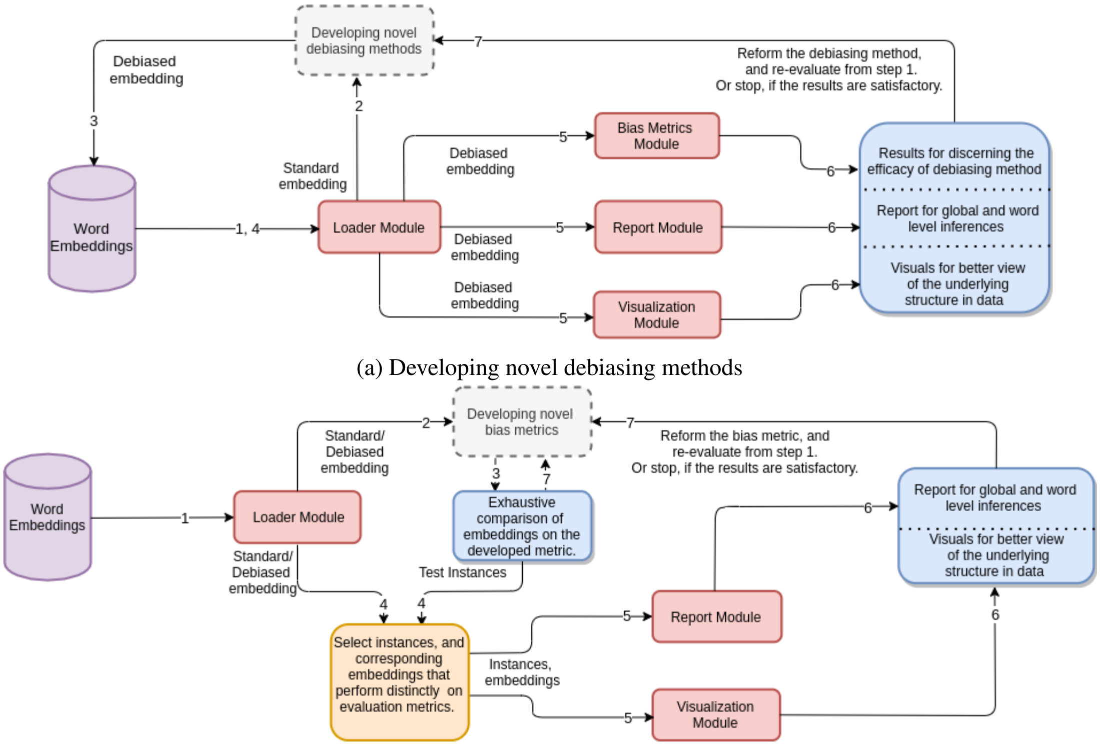
3.3 Bias Metrics Module

Motivation: Evaluation metrics provide the necessary quantitative support for comparing and contrasting between different debiasing methods. However, different research articles often show different results for the same metric despite having theoretically similar configurations. The bias metrics module of FEE is aimed at filling this gap, it provides a suite of bias metrics built on a common framework for facilitating reliable inference.

Working: The bias metrics module of FEE currently provides access to a number of evaluation metrics, as shown in Figure 1. Each evaluation metric is represented by a unique class in the module. Each metric class depends on some common utilities and consists of multiple methods that implement their unique evaluation procedure. Similar to the debiasing module, each metric class's instance is initialised by a `WE` object. The metrics either operate on a single word, pair of words or list of words. Each metric class has a common method called `compute()` that returns the final result by accepting different arguments corresponding to the type of metric. The unified design of bias metric module fosters a standardized access to any bias based evaluation metric and facilitates reproducible research.

3.4 Visualization Module

Motivation: Visualizations provide useful insights into the behaviour of a set of data points. Many prior debiasing methods (Bolukbasi et al., 2016a; Kumar et al., 2020) have strongly motivated their work by illustrating certain undesirable associations prevalent in standard word embeddings. Thus,


Developing novel debiasing methods
Debiased embedding
7
Reform the debiasing method, and re-evaluate from step 1. Or stop, if the results are satisfactory.
3
2
Standard embedding
Word Embeddings
1, 4
Loader Module
Debiased embedding
5
Bias Metrics Module
6
Results for discerning the efficacy of debiasing method
Report for global and word level inferences
Visuals for better view of the underlying structure in data
Debiased embedding
Report Module
Debiased embedding
Visualization Module
Developing novel bias metrics
Standard/ Debiased embedding
2
7
Reform the bias metric, and re-evaluate from step 1. Or stop, if the results are satisfactory.
Word Embeddings
1
Loader Module
3
Exhaustive comparison of embeddings on the developed metric.
Report for global and word level inferences
Visuals for better view of the underlying structure in data
Standard/ Debiased embedding
Test Instances
4
Select instances, and corresponding embeddings that perform distinctly on evaluation metrics.
Instances, embeddings
Report Module
Visualization Module
5
6

(a) Developing novel debiasing methods

(b) Developing novel bias metrics

Figure 2: FEE serves as a centralized resource for practitioners and researchers to develop novel debiasing methods and bias evaluation metrics. Figure (a) and (b) illustrate the possible workflow associated with each of the tasks respectively all made possible by the powerful abstraction provided by FEE.

through `FEE` we also provide off the shelf visualization capabilities that might help users to build reliable intuitions and uncover hidden biases in their models.

Working: In this module, we implement a separate class for each visualization type. Just like other modules, a visualization class object is initialized by `WE` object, and makes use of some common utilities. Each visualization class has a `run()` method that takes in a word list and other optional arguments for producing the final visualizations. Figure 1 illustrates some off the shelf visualization options provided by FEE.

3.5 Report Module

Motivation: The bias metrics and visualization modules incorporate a plethora of components which provide an exhaustive set of results. However, sometimes a specific combination of their components can provide the needed information succinctly. Accordingly, the report module aims to provide a descriptive summary of bias in word embeddings at the word and global level.

Working: The report module is comprised of two separate classes that are representative of a word and a global level report respectively. Both the classes operate on `WE` initialized embedding object and implement a common `generate()` method that creates a descriptive report. The `WordReport` class is useful for providing an abridged information about a single word vector in terms of bias. A call to the `generate()` method of `WordReport` utilizes the components of other modules and instantly reports the direct bias, proximity bias, neighbour analysis (`NeighboursAnalysis`), neighbour plot and a neighbour word cloud for a word. The `GlobalReport` class, in contrast creates a concise report at the entire embedding level. Unlike the word level, `GlobalReport` class does not make use of the other modules, since it achieves all the required content from the embedding object. The `generate()` method of `WordReport` provides the information about n most and least biased words in a word embedding space.

4 Developing new methods with FEE

Despite the development of a large number of debiasing methods, the issue of bias in word representations still persists (Gonen and Goldberg, 2019) making it an active area of research. We believe that the design and wide variety of tools provided by FEE can play a significant role in assisting practitioners and researchers to develop better debiasing and evaluation methods. Figure 2 portrays FEE assisted workflows which abstract the routing engineering tasks and allow users to invest more time on the intellectually demanding questions.

5 Conclusion and future work

In this paper, we described Fair Embedding Engine (FEE), a python library which provides central access to the state-of-the-art techniques for quantifying, mitigating and visualizing gender bias in non-contextual word embedding models. We believe that FEE will facilitate the development and testing of debiasing methods for word embeddings. Further, it will make it easier to visualize the existing bias present in word vectors. In future, we would like to expand the capabilities of FEE towards contextual word vectors and also provide support towards biases other than gender and language other than English. We also look forward to integrate OSS such as WEFE (Badilla et al., 2020) to enhance the bias evaluation capabilities of FEE.

References

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.

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. 2016a. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016b. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.

Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. *CoRR*, abs/1904.03035.

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

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

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):E3635–E3644.

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. *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)*, page 609–614.

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

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 1641–1650.

Vaibhav Kumar, Tenzin Singhay Bhotia, Vaibhav Kumar, and Tanmoy Chakraborty. 2020. Nurse is closer to woman than surgeon? mitigating gender-biased proximities in word embeddings. *Transactions of the Association for Computational Linguistics*, 8:486–503.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2018. Gender bias in neural natural language processing. *arXiv preprint arXiv:1807.11714*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013,Workshop Track Proceedings*, pages 1–12.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Travis E Oliphant. 2006. *A guide to NumPy*, volume 1. Trelgol Publishing USA.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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.

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.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. 2011. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22.

Zekun Yang and Juan Feng. 2020. A causal inference method for reducing gender bias in word embedding relations. In *AAAI*, pages 9434–9441.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018b. Learning gender-neutral word embeddings. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, page 4847–4853.