Applied Soft Computing 133 (2023) 109916

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Multi-objective search for gender-fair and semantically correct word embeddings

Max Hort, Rebecca Moussa, Federica Sarro [*]

Department of Computer Science, University College London, London, WC1E 6BT, United Kingdom

ARTICLE INFO

Article history:
Received 4 March 2022
Received in revised form 28 October 2022
Accepted 2 December 2022
Available online 9 December 2022

Keywords:
Software fairness
Search-based software engineering
Gender bias
Word embeddings

ABSTRACT

Fairness is a crucial non-functional requirement of modern software systems that rely on the use of Artificial Intelligence (AI) to make decisions regarding our daily lives in application domains such as justice, healthcare and education. In fact, these algorithms can exhibit unwanted discriminatory behaviours that create unfair outcomes when the software is used, such as giving privilege to one group of users over another (e.g., males vs. females). Mitigating algorithmic bias during the development life cycle of AI-enabled software is crucial given that any bias in these algorithms is inherited by the software systems using them. However, previous work has shown that mitigating bias can impact the performance of such systems. Therefore, we propose herein a novel use of soft computing for improving AI-enabled software fairness. Specifically, we exploit multi-objective search, as opposed to previous work optimising fairness only, to strike an optimal balance between reducing gender bias and improving semantic correctness of word embedding models, which are at the core of many AI-enabled systems. To assess the effectiveness of our proposal, we carry out a thorough empirical study based on the most recent best practice for the evaluation of search-based approaches and AI-enabled software. We explore seven different search-based approaches, and benchmark them against both baseline and state-of-the-art approaches applied to a popular and widely used word embedding model, namely Word2Vec. Our results show that multi-objective search outperforms single-objective search, and generates word embeddings that are strictly better than the original ones in both objectives, bias and semantic correctness, for all investigated cases. Additionally, our approach generates word embeddings of higher semantic correctness than those generated by using state-of-the-art techniques in all cases, while also achieving a higher degree of fairness in 67% of the cases. These findings show the feasibility and effectiveness of multi-objective search as a tool for engineers to incorporate fair and accurate word embedding models in their AI-enabled systems.

1. Introduction

Fairness in software systems aims to provide algorithms that operate in a non-discriminatory manner [1], with respect to protected attributes such as gender, race, or age.[1]

Ensuring fairness is a crucial non-functional property of modern software systems [2–6], especially those that rely on Artificial Intelligence (AI) and Machine Learning (ML) algorithms to make decisions that can dramatically affect peoples' lives such as criminal justice [7,8], finance [9], and recruitment [10]. For example, it has been found that software systems used for recidivism assessment in justice courts are more likely to falsely label black defendants as future criminals at almost twice the rate as white defendants [7]. Also, software systems used by companies for job advertisement and recruitment have often shown gender bias against women [10,11].

In this paper, we propose a novel use of soft computing to mitigate gender bias in word embedding models.

Word embeddings have rapidly become an all-purpose tool serving a variety of day-to-day tasks (e.g., item recommendations [12,13], spam detection [14] and web search [15,16]), as well as Software Engineering (SE) tasks, such as sentiment analysis [17], bug report recommendation [18] and information retrieval for API documents [19].

A word embedding model is a representation of words trained from unannotated text corpora, where words with a similar meaning have a similar vector representation. Training word embedding models does not only require a large amount of data, which is often time consuming and resource expensive, but, just like any model that requires training, these models are also prone to inherit possible stereotypical social biases present

[*] Corresponding author.
E-mail address: f.sarro@ucl.ac.uk (F. Sarro).

[1] Protected attributes are those qualities, traits or characteristics that, by law, cannot be discriminated against.

https://doi.org/10.1016/j.asoc.2022.109916

in the training corpora [20,21]. Blindly using pre-trained word embedding models, without considering underlying biases, can lead to problems, which have been already detected in various software applications, such as multi-label classification [22] and co-reference resolution [23]. The problem is exacerbated by the wide adoption of open-source word embedding models pre-trained on vast corpora, such as news [24] and encyclopedia [25]. These are often quite easy and inexpensive to use, which leads to any underlying biases to quickly spread across a wide range of software applications. For example, Bolukbasi et al. [20] showed that a popular word embedding model pre-trained on news articles exhibited a bias towards gender, as it learned the analogy that "man to computer programmer" is the same as "woman to homemaker". It is obvious to see why it would be problematic to use such a model for information retrieval in job application processes, as male names, which are closer to $\overrightarrow{man}$ than $\overrightarrow{woman}$, would have a higher similarity to the job of a computer programmer than female names would. Thus, it is of great importance to remove or at least mitigate any existing bias in pre-trained word embedding models before using them. Nonetheless, achieving fairness has a cost [26]. Bias mitigation approaches can damage the performance of a machine learner while making it fair. This is known as the accuracy-fairness trade-off.

While various techniques have been proposed to mitigate gender bias in word embeddings [20,22,27,28], they have all focused on reducing bias only. Therefore, we propose the use of a novel multi-objective soft computing approach to reduce gender bias while simultaneously improving the semantic correctness of word embeddings. Specifically, our proposal is based on the use of search-based approaches to automatically adapt pre-trained word embedding models to strike an optimal trade-off between gender bias and semantic correctness, whereas previous work only sought to reduce gender bias [20,29–31].

To assess the effectiveness of our proposal, we design and experiment with several local (i.e., single state-based) and global (i.e., population-based) search techniques (e.g., Tabu Search, Hill Climbing, Genetic Algorithms) to optimise a popular and widely used word embedding model, namely Word2Vec (W2V), pre-trained on Google news articles [24], with six different pairs of train–test data based on two publicly available and widely-used datasets (namely, WEATs [21] and MEN [32]). We benchmark our approach with the original pre-trained W2V model and state-of-the-art debiasing approaches (i.e., Hard Debiasing [20] and Linear Projection [29]).

Our findings show that both, local and global single-objective search approaches optimising word embeddings for gender bias solely, produce, on average, models that are less biased than the original pre-trained ones in 67% of the cases (*p-value* <0.01). However, such a notable improvement in fairness comes at the cost of reducing the semantic correctness of the models in 89% of the cases. On the other hand, we find that using multi-objective search not only allows us to prevent such a detrimental effect but also produces word embeddings that are always strictly better than the original ones in both objectives, bias and correctness. In particular, both the use of a single weighted function and the use of a Pareto-optimal approach, lead to solutions with a significantly better bias vs. correctness trade-off than those of the original pre-trained models. They are also able to improve their semantic correctness in all cases, as opposed to the state-of-the-art debiasing models, while also achieving a higher fairness than the state-of-the-art in 67% of the cases.

Our results suggest that our approach can be adopted by engineers, who rely on language models in their software systems, in order to automatically incorporate fairness into the development of AI-enabled systems based on word embeddings. Additionally, since we apply multi-objective optimisation, fairness does not come at the cost of correctness of the language model and, therefore, the engineers can create fairer software without the downside of sacrificing performance by using the approach we propose herein.

To summarise, the main contributions of our paper are:

- the formulation of the pre-trained word embedding adaptation and debiasing problem as a search-based problem;
- the proposal of a multi-objective evolutionary genetic algorithm to reduce gender bias and increase semantic correctness, simultaneously;
- a thorough empirical study to evaluate the effectiveness of our proposal by benchmarking it with random search, local and global search (either single- and multi-objective), and state-of-the-art bias mitigation approaches [20,29], in order to adapt a popular pre-trained word embedding model (i.e., W2V [24]) on six different training–testing datasets, which are publicly available and widely-used in the literature [21,32].

We also make the scripts and data used in our study publicly available online [33] to allow for replication and extension of our work.

The rest of the paper is organised as follows. Section 2 discusses related work on software fairness, and the adaptation and debiasing of word embeddings. Section 3 presents the word embedding adaptation problem as a search-based problem, and our proposed approach to tackle it. The design of our empirical study is described in Section 4 and its results are discussed in Section 5. Section 6 presents conclusions and future work.

2. Related work

2.1. Realising fair software

Software fairness is a growing concern for those engineers realising AI-enabled software. In their FSE'18 vision paper, Brun and Meliou [34] advocate for novel strategies to achieve fairness during the development life-cycle of such systems. In fact, for software systems that rely on AI, the design of fairness-aware algorithms that can produce fair models is critical [1,34], as these models are widely used in software systems.

This is a challenging task as fairness often comes at the cost of other important properties such as classifier accuracy or model correctness [35–37]. Therefore, recent work has explored the power of multi-objective optimisation to account for these multiple competing objectives. We focus below on the description of such work and the comparison with ours, whereas we refer the reader to existing surveys on algorithmic bias mitigation methods [6,38–40].

The FSE'20 work by Chakraborty et al. [41] has shown how to integrate bias mitigation into the design of classification ML models. Specifically, they proposed a multi-objective approach for hyperparameter tuning of Logistic Regression to optimise for both fairness and accuracy. Their results show that one can achieve fairness without highly reducing the accuracy of the classification model.

This work inspired us to investigate the use of multi-objective optimisation to integrate bias mitigation into the software development life-cycle of Natural Language Processing (NLP) methods such as word embeddings. The results of our empirical study (see Section 5) show that our proposed approach is able to optimise both, fairness and correctness, of pre-trained word embeddings. Therefore, suggesting that multi-objective search is suitable not only to optimise the fairness of traditional classification approaches [41], but also those of more advanced NLP ones like word embeddings.

A different use of multi-objective optimisation to address software fairness, can be found in the software requirements domain. Finkelstein et al. [42] have been the first to utilise multi-objective optimisation to mitigate bias in user requirements prioritisation when realising software systems. For example, some customers may wish to receive equal spend from the developers, while others may prefer to receive an equal number of their desired requirements compared to other customers.

Some empirical studies have also been carried out to gain more insights on the trade-off between software fairness and performance of AI-enabled systems. Hort et al. [43] proposed Fairea, a benchmarking approach for comparing the effectiveness of bias mitigation methods for binary classification, which takes into account and aims at quantifying the trade-off between accuracy and fairness. Biswas and Rajan [44] carried out a large empirical evaluation of fairness and mitigations of real-world crowd-sourced ML models. They pointed out that trade-offs between accuracy and fairness measures exist and should be considered when deploying bias mitigation methods. To improve the fairness-performance trade-off for ML models, Chen et al. [45] used an ensemble approach, which combined models trained for different objectives (i.e., fairness and performance metrics). Hort and Sarro [46] showed that while bias of ML models can be reduced, this can come at the cost of losing the ability to differentiate between desired features.

2.2. Adapting word embeddings

Pre-trained word embeddings are the embeddings learned in a given task but which can be used for solving another task. Research has been conducted on the evaluation and improvement of such pre-trained word embeddings at a post-processing stage, in order to improve their correctness when used in a domain different from the one they were trained for. This is often referred to as word embedding adaptation. The majority of the adaptation methods are based on counter-fitting [47–50], an adaptation method proposed by Mrkšić et al. [47]. Other adaptation approaches include manifold learning, graph-based techniques, etc. [51–57]. Our proposed approach is different from previous ones as they aim at adapting pre-trained word embeddings to different domains, therefore only improving their correctness, while our approach aims at adapting pre-trained word embeddings to simultaneously enhance their fairness and correctness by exploiting the power of multi-objective search.

2.3. Debiasing word embeddings

In order to improve algorithm fairness, three types of approaches can be applied: pre-processing, in-processing and post-processing. We refer the reader to the literature review by Sun et al. [58] for further details on techniques including the debiasing of training corpora (pre-processing) and debiasing by adjusting algorithms (in-processing). In the following, we focus on post-processing debiasing methods for word embeddings as our work belongs to this category.

Post-processing bias methods mitigate bias after a model has been trained. To reduce gender bias, Bolukbasi et al. [20] proposed the following Hard Debiasing (HD) post-processing method: They first identified a gender subspace and then proceeded to neutralise it, ensuring that the performance on the evaluation tasks is maintained. To determine the gender subspace, they identified a gender direction $g \in \mathbb{R}^d$ by combining several directions (e.g., $\overrightarrow{she} - \overrightarrow{he}$, $\overrightarrow{woman} - \overrightarrow{man}$). This approach has however been criticised to have several shortcomings [59] and alternative approaches have been sought to overcome them. To this end, Dev and Phillips [29] debiased word embeddings by using a linear projection along the gender direction, whereas, Lauscher et al. [31] adjusted the linear projection by using an alternative projecting approach and a neural network to learn word vectors transformation. Instead, Kaneko and Bollegala [60] proposed the use of an autoencoder to remove biases from pre-trained word embeddings. Shin et al. [61] proposed a latent disentanglement method to obtain gender-neutralised word embeddings. Ravfogel et al. [30] presented an Iterative Null-space Projection (INLP) method. Instead of specifying a gender direction, as done by Bolukbasi et al. [20], INLP learns this direction with a linear classifier and removes it by iteratively projecting the word embeddings on their null-space. Kaneko and Bollegala [62] debiased pre-trained word embeddings with the use of dictionaries.

In this work, we take a different route to mitigate bias in word embeddings. Unlike the proposals described above, which checked for potential degradation in the semantic correctness of word embeddings after debiasing [20,29–31,61], our approach is the first to incorporate semantic correctness in the debiasing procedure. Moreover, we aim to reduce bias by adapting all vectors in their entirety instead of focusing the adaptation on particular vector components. Existing approaches have mainly sought to debias word embeddings by removing biased portions of the data or the gender direction in pre-trained embeddings. Whereas, our work is the first to formulate word embedding adaptation and debiasing as an optimisation problem and to propose the use of multi-objective search-based approaches in order to automatically optimise the original pre-trained model by simultaneously minimising its bias and maximising its semantic correctness. Such search techniques have the advantage of being applied to various optimisation problems (e.g., are not restricted to convex solution spaces). Moreover, unlike previously proposed techniques, the multi-objective nature of our proposed approach, which explores various trade-offs and allows for both bias and correctness to be optimised simultaneously, gives the engineers the advantage of choosing the most suitable solution according to the problem they are tackling (i.e., more importance to correctness or fairness). The results of our empirical study show that using a multi-objective approach does not only reduce bias but also improves the semantic correctness of the original model, as opposed to the results achieved in previous work which only reduces bias.

3. Mitigating word embedding bias as a search-based problem

In this section we explain our proposal to formulate the problem of adapting and debiasing pre-trained word embeddings as an optimisation search-based problem, where the pre-trained model can be iteratively optimised aiming at minimising bias (single-objective formulation) [55], or at minimising bias and also maximising its correctness (bi-objective formulation). Both local and global search, can be applied to tackle either the single- and bi-objective problem formulation.

3.1. Word embedding models

Given a text corpora, word embedding models are trained on co-occurrences of words in an underlying text in order to learn semantically similar words appearing in similar contexts. Word embedding models represent words w as vectors of dimensionality d: $\overrightarrow{w} \epsilon \mathbb{R}^d$.[2] Furthermore, they can be used to display and investigate relationships of words [63] as in the example below, which shows that the male/female relationship is learned and

[2] The dimension of a vector space V is the cardinality (i.e. the number of vectors) of a basis of V over its base field.

models are able to represent the analogy between "man-king" and "woman-queen":

$$\overrightarrow{man} - \overrightarrow{woman} \approx \overrightarrow{king} - \overrightarrow{queen} \quad (1)$$

Different pre-trained word embedding models can be devised based on different approaches and corpora [24,25,64]. Among the most popular, we find the one by Mikolov et al. [24], who built a W2V model pre-trained on Google news articles, the GloVe ones [25], which were trained on corpora from different domains, including Twitter, Wikipedia or Common Crawl,[3] and fastText [64], which provides pre-trained models for 157 languages.[4]

3.2. Solution representation

After training a word embedding model on a given text corpora, the semantically similar words appearing in similar contexts will be represented by similar word vectors. Given that our goal is to de-bias such pre-trained models, we aim at changing these vectors to adjust for unfair similarity values learnt from biased data. Thus, a solution to our problem $\overrightarrow{s}$ is a vector of real numbers of the same length as the vector of the original pre-trained model $\overrightarrow{w}$ (e.g., in the empirical study presented in Section 4, $\overrightarrow{s}$ is a vector of size 300 as the original pre-trained W2V vector length is 300). Such a vector will be used to modify each of the existing word vectors $\overrightarrow{w}$ in order to recompute and adapt the word vectors constituting the original pre-trained word embedding. This results in an adapted set able to minimise bias and maximise accuracy (according to the fitness function explained in Section 3.4). Specifically, vector multiplication between $\overrightarrow{s}$ and $\overrightarrow{w}$ is used to this end. A word vector $\overrightarrow{w}$ is multiplied by a solution vector $\overrightarrow{s}$, of the same size, element by element. The result is a modified word vector $\overrightarrow{w'} = \overrightarrow{w} \circ \overrightarrow{s}$. This procedure is applied to every word vector of a word embedding model.

3.3. Initialisation and neighbour creation

The first step of any search algorithm is to randomly generate an initial solution from a set of possible solutions. One can start with a single solution (local search, also known as single state-based search) or multiple solutions (global search, also known as population-based search).

An initial solution $\overrightarrow{s}$ to our problem is obtained by adding a small uniform noise vector ($\overrightarrow{noise}$) to a vector of all ones ($\overrightarrow{1}$) as follows:

$$\overrightarrow{s} = \overrightarrow{1} + \overrightarrow{noise} \quad (2)$$

We use $\overrightarrow{s} = \overrightarrow{1}$ as a base vector because a multiplication by a vector of ones' does not warrant any changes. In case of a set of n solutions, the above step is repeated n times in order to initialise each of the solutions in the set.

The next step is to evolve such initial solution(s) towards better ones. To this end, "neighbourhood" solutions are iteratively created, and the "goodness" of each solution is evaluated according to one or more objective (i.e., fitness) functions. As vectors have continuous values, it is not feasible to generate and explore all possible neighbours to search for the best one. Therefore, we consider and evaluate different strategies for neighbour creation, as follows:

1. *Noise value*: adding a small noise value to a single element of $\overrightarrow{s}$ to create a new neighbour solution $\overrightarrow{s_n}$;
2. *Noise vector*: adding a small uniform noise vector, in the range $[-0.02, 0.02]$ to $\overrightarrow{s}$ to create a new neighbour solution $\overrightarrow{s_n}$;
3. *Inversion*: inverting the change of an element in contrast to $\overrightarrow{1}$ (e.g., $1.05- > 0.95$ $(1 + (1 - 1.05))$ to create a new neighbour solution $\overrightarrow{s_n}$;
4. *Swap*: Swapping two elements of $\overrightarrow{s}$ to create a new neighbour solution $\overrightarrow{s_n}$.

In our empirical study (see Section 4–5), we experiment with each of the above methods for neighbour creation. In Section 4.5, we describe the method that was used in combination with the search algorithms investigated in our study.

3.4. Fitness function

The fitness function determines how fit (i.e., good) a candidate solution is for the problem at hand. Such a function is applied by assigning a score to each solution. The probability that a solution will be selected for the subsequent iteration is based on its score (i.e., the score measures the ability of a solution to compete with others).

In this work we are interested in minimising the gender bias of word embedding models and maximising their semantic correctness. Therefore, given a solution $\overrightarrow{s}$ to our problem, we need to define two fitness functions: one to compute its gender bias and one to compute its semantic correctness.

In order to evaluate gender bias of a word embedding model, we define the *bias* fitness function based on the Word Embedding Association Tests (WEATs) [21] (Eq. (3)), which analyse the similarity of two sets of target words with two sets of attribute words.

We adapt the WEAT test statistic t to compute the *bias* [21] as follows:

$$bias(\overrightarrow{x}) = t(T, A, B) = \sum_{x \epsilon T} |s_w(x, A, B)| \quad (3)$$

$$s_w(w, A, B) = mean_{a \epsilon A} cos(\overrightarrow{w}, \overrightarrow{a}) - mean_{b \epsilon B} cos(\overrightarrow{w}, \overrightarrow{b}) \quad (4)$$

where $T = X \bigcup Y$ is a union of both target sets. A and B are attribute sets of identical size. $t(T, A, B)$ computes the test statistic and $s_w(w, A, B)$ calculates the difference in similarity of attribute sets to a word w. As a result, the vectors $\overrightarrow{w}$, $\overrightarrow{a}$, $\overrightarrow{b}$ are obtained from the same word embedding model and have the dimensionality d ($\mathbb{R}^d$). For example, the W2V model pre-trained on news articles exhibits the following similarity (according to cosine similarity), between the words Amy and John, and the target attribute family:

```
sim(Amy, family) = 0.17
sim(John, family) = 0.09
```

which is clearly biased towards gender since a female name (i.e., Amy) has a higher similarity to the attribute family than a male name (i.e., John) has. On the other hand, using our proposed approach, we are able to generate a de-biased model showing balanced similarities:

```
sim(Amy, family) = 0.09
sim(John, family) = 0.08
```

Further details on the WEAT datasets used in our empirical study are provided in Section 4.3.

In order to evaluate the semantic correctness of word embeddings, we use a *semantic* fitness function (Eq. (5)) based on the word similarity method [65], which is widely used in previous work [20,29,53]. This is an intrinsic evaluation method, where the semantic score is calculated based on the correlation of human judged similarity with the one predicted by the word embedding model. In other words, given a list of word pairs, humans determine their similarity (e.g., "bread" and "chair" receive a similarity

[3] These models are available at https://nlp.stanford.edu/projects/glove/

[4] https://fasttext.cc/

score of 0.14 on a scale from 0–1, with 1 indicating identical). Then, word embedding models are used to compute similarities (using cosine similarity) on the same list of word pairs. The more the word embedding results resemble human judgement, the better. To compute the correlation between word embedding results and human judgment, we used the Spearman's ρ rank correlation coefficient [66] as done in previous work [55,65,67,68], and the MEN data [32]. The definition of the semantic fitness function is as follows:

$$semantic(\overrightarrow{x}) = \rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{5}$$

where d is the pairwise distances of the ranks of the word pairs according to MEN and the word embedding model; n is the number of samples. The ρ correlation coefficient ranges between -1 and 1, which indicates a perfect inverse and direct correlation, respectively. Further details on the MEN data used in our empirical study are provided in Section 4.4

3.5. Handling multiple objectives

Given that one of our goals is to investigate whether we could use search-based approaches to simultaneously optimise both semantic correctness (Eq. (5)) and gender bias (Eq. (3)), we combine these two objectives into a single weighted fitness function, as follows:

$$f(\overrightarrow{x}) = w_1 * semantic(\overrightarrow{x}) - w_2 * bias(\overrightarrow{x}) \tag{6}$$

where w_1, w_2 are adjustable weights that sum up to 1 and $\overrightarrow{x}$ is a solution vector. Solutions aim at maximising semantic correctness while minimising gender bias.

Another approach to handle multiple objectives is to measure them on orthogonal scales and to evaluate them for *Pareto-optimality* [69], which states that "a solution x_1 dominates another solution x_2 if it is not worse in all objectives than x_2 and better in at least one". Any multi-objective evolutionary algorithm, such as the NSGA-II algorithm [70], can be used to rank solutions based on Pareto-optimality.

The first approach, which we call Weighted Sum Method, generates a single optimal solution, while approaches based on Pareto-optimality produce a set of equally viable, non-dominated solutions. In our paper we investigate and compare the effectiveness of both approaches to simultaneously optimise the two objectives of improving semantic correctness and reducing gender bias.

4. Empirical study design

In this section, we outline the design of the experiments we carry out to investigate the effectiveness of search-based approaches for optimising word embedding models with regards to gender bias and semantic correctness.

4.1. Research questions

In order to evaluate whether search-based approaches are able to optimise and adapt pre-trained word embedding models, we first investigate their ability to minimise gender bias, which motivates our first research question:

RQ1. Single-Objective Optimisation: Are search-based approaches able to reduce word embeddings gender bias?

To answer this question, we investigate the ability of three single objective search-based approaches (i.e., Hill Climbing, Tabu Search and Genetic Algorithm) to reduce the gender bias of a popular pre-trained word embedding model (i.e., W2V). A description of W2V is given in Section 4.2, while the search algorithms and their settings are described in Section 4.5. As our approach is the first to use meta-heuristic search to debias pre-trained word embedding models, we benchmark it against both the original pre-trained word embedding model and a word embedding model obtained via Random Search. We compute the gender bias of the original pre-trained W2V model as a baseline (we refer to it as *Base* from now on) as we expect search algorithms to at least maintain the original performances. We use Random Search (RS) as it is the recommended baseline for search-based approaches when there is no comparable state-of-the-art for the problem at hand [71–74], as in our case.

Gender bias is an important aspect of word embeddings but it is not the only one. The semantic correctness of these models is crucial to guarantee meaningful semantic structure in the respective vector spaces [75]. Therefore, even if we find that search-based approaches are able to minimise gender bias, we need to carefully check whether this negatively impacts their semantic correctness. Indeed, previous work, which proved the effectiveness of single-objective search for various software engineering tasks, has also observed a detrimental effect on other objectives of interest, including software fairness (see e.g., [41, 76]). This motivates our second research question:

RQ2. Detrimental Effects: Does optimising gender bias reduce the semantic correctness of word embeddings?

To answer this question, we investigate whether the semantic correctness of the models produced in RQ1 differs from the correctness of the original pre-trained word embedding model.

Since single-objective search-based approaches for reducing gender bias might negatively affect semantic correctness, our third and last research question investigates the use of multi-objective approaches, which are designed to find optimal trade-offs among multiple competing objectives. In particular, we aim to assess whether simultaneously optimising measures of fairness and semantic correctness allows us to reduce gender bias while preserving (and possibly improving) semantic correctness:

RQ3. Multi-objective Optimisation: Are multi-objective search-based approaches able to optimise word embeddings for both gender bias and semantic correctness?

To answer this question, we investigate two widely-known approaches in multi-objective search. The first, named Weighted Sum Method (WSM), which consists of combining two or more objectives into a single fitness function, and using this function to guide single-objective search methods. In particular, we use the same search methods investigated in RQs 1–2 (i.e., HC, TS, GA) guided by a weighted fitness function combining gender bias and semantic correctness as per Eq. (6), and experiment with 11 different weights (see Section 4.5). In the second approach, we examine the simultaneous optimisation of multiple objectives based on the concept of Pareto-optimality [69]. In particular, we use a popular multi-objective evolutionary algorithm, namely NSGA-II [70], described in Section 4.5. We investigate both approaches as the weighted sum method is generally quicker to execute than the approaches based on Pareto-optimality. However, the weighted fitness function needs to be designed carefully as the results may depend on the weighted combination chosen. To answer RQ3, we benchmark the effectiveness of multi-objective approaches with respect to both, the same baselines and single-objective search algorithms used in RQs 1–2. Additionally, we compare our approaches with two state-of-the-art bias mitigation methods for word embeddings: Hard Debiasing (HD) by Bolukbasi et al. [20] and Linear Projection (LP) by Dev and Phillips [29] as they are the most representative and their implementation is publicly available.[5] Moreover, in Section 5.4, we provide information on the runtime and space complexity of the search approaches.

[5] HD is the most used and well known de-bias method (with 1.4k cites), while LP was subsequently proposed by Dev and Phillips aiming at reducing

Table 1
Semantic correctness and gender bias of the original pre-trained word embeddings model W2V and its debiased version obtained by using the HD and LP state-of-the-art methods for each of the dataset used in our study. Better solutions have higher values of semantic correctness and lower values of gender bias.

Measure (dataset)	W2V [24]	HD [20]	LP [29]
Semantic correctness (MEN)	0.77	0.77	0.77
Gender bias (W6)	1.25	0.75	1.10
Gender bias (W7)	0.47	0.10	0.48
Gender bias (W8)	0.54	0.09	0.50

4.2. Pre-trained word embedding model

We investigate a popular and publicly available pre-trained word embedding model (i.e., W2V [24]) which computes word embeddings based on local context information. We use a W2V model trained on Google news articles.[6] The Google news W2V model provides 300-dimensional word vectors with continuous values, and contains word vectors of 3 million words in total. Table 1 shows the semantic correctness and gender bias values achieved by the original pre-trained W2V model as well as the values obtained by debiasing it with the state-of-the-art methods HD [20] and LP [29], which are used as a benchmark in our empirical study.

4.3. WEATs data for gender bias

As explained in Section 3.4, in order to evaluate the gender bias of a word embedding model, we define the *bias* fitness function based on WEATs [21], as per Eq. (3). The goal of WEATs is to determine whether there is a difference between attribute words in regards to their similarity to target words [21]. We use all the available WEATs pairs of target word sets and attribute word sets related to gender as provided by Caliskan et al. [21]: WEAT 6 (W6) uses male and female target words with career and family attributes; WEAT 7 (W7) and WEAT 8 (W8) both use male and female attributes, with math and arts (W7), and science and arts (W8) target words.[7] The target (X, Y) and attribute (A, B) words of WEAT 6, 7 and 8 are as follows:

WEAT 6:
X = {John, Paul, Mike, Kevin, Steve, Greg, Jeff, Bill}
Y = {Amy, Joan, Lisa, Sarah, Diana, Kate, Ann, Donna}
A = {executive, management, professional, corporation, salary, office, business, career}
B = {home, parents, children, family, cousins, marriage, wedding, relatives}

WEAT 7:
X = {math, algebra, geometry, calculus, equations, computation, numbers, addition}
Y = {poetry, art, dance, literature, novel, symphony, drama, sculpture}
A = {male, man, boy, brother, he, him, his, son}
B = {female, woman, girl, sister, she, her, hers, daughter}

WEAT 8:
X = {science, technology, physics, chemistry, Einstein, NASA, experiment, astronomy}
Y = {poetry, art, Shakespeare, dance, literature, novel, symphony, drama}
A = {brother, father, uncle, grandfather, son, he, his, him}
B = {sister, mother, aunt, grandmother, daughter, she, hers, her}

4.4. MEN data for semantic correctness

We use the MEN dataset [32] to compute the semantic correctness with the word similarity method [55,67,79,80] after a careful analysis of the public datasets available. Indeed, only two datasets in the literature (namely, MEN and SimVerb-3500) have a predefined train–test split, however SimVerb-3500 only contains verbs thus limiting its usage. The MEN dataset is also one of the biggest in terms of number of word pairs. We use the version 0.2, released on 30/04/2012.[8] In the MEN dataset, the word pairs are denoted in the format word1-wordtype word2-wordtype similarity, for example: ivy-n plant-n 45.000000 is a word pair. We normalise similarity to a range of [0, 1], as suggested in previous work [55].

4.5. Computational search

In this section we describe the search-based algorithms investigated to answer our research questions.

Random Search (RS) is usually the most naive search one could think of for the problem at hand, and it is used to benchmark more advanced search strategies [81]. In our study, RS generates solutions by adding random noise vectors to the unit vector. We experiment with different levels of uniform noise, ranging from 0.05 − 0.5 with a step size of 0.05, over 10,000 repetitions. As the results showed that a noise level of 0.05 achieves the best performance, we set to this value the level of noise of an initial solution in our experiments.

Hill Climbing (HC) is a stochastic local search algorithm, which starts with a random solution and evolves it by exploring one neighbour at time created by using *noise values* (as described in Section 3.3). The current solution is updated, at each iteration, only if the neighbour is considered better/more accurate [69], otherwise the search stops. We experiment HC with levels of noise between 0.02 − 0.2 with a step size of 0.02 and the best results were found with a noise level of 0.14.

Tabu Search (TS) is a heuristic search algorithm that can be used to augment other heuristics [82]. It starts from a randomly created solution, and it explores a set of neighbours at each iteration. The current solution is update only if a better one is found, and the search ends when a stopping criterion is met (e.g., a maximum number of iterations is reached). TS also makes use of a *Tabu List* containing solutions that should not be explored by the search algorithm in subsequent iterations. TS uses the same *noise values* as RS and HC to create a neighbouring solution (Section 3.3). We experiment TS with different levels of noise (between 0.02 − 0.2 with a step size of 0.02 as done for HC) and different Tabu List sizes (namely $\{5, 10, 25, 50, 75, 100, 150\}$). We observe that the best performance is achieved with a noise

HD's shortcoming as described in our Related Work Section 2.3. HD is available at https://github.com/tolga-b/debiaswe and LP is available at https://github.com/sunipa/Attenuating-Bias-in-Word-Vec

[6] https://code.google.com/archive/p/word2vec/

[7] WEATs contain 10 datasets [21]: seven concern different type of bias, while three concern a range of topics, such as insects and flowers, which are not relevant to software system fairness. Since our work focuses on gender bias, we used all the three WEATs datasets relevant to gender [77,78] in our analysis, and include them in our on-line appendix [33]. The remaining four datasets describe age and race bias, which can be explored with multi-/many-objective approaches in future work.

[8] MEN is publicly available at https://staff.fnwi.uva.nl/e.bruni/MEN. The W2V model does not contain the words {*colour*, *grey*, *harbour*, *theatre*}, so we used {*color*, *gray*, *harbor*, *theater*} to maintain the same dataset size [55].

level of 0.14 and a Tabu List size of 150. These values are therefore used for neighbour creation in our experiments.

Genetic Algorithm (GA) is a global search technique inspired by the Darwinian theory of evolution [83]. At each generation, GA applies operators such as crossover and mutation which allows it to combine and exchange characteristics of selected solutions [84]. GA mimics the natural selection procedure whereby fitter solutions have a higher chance of being selected and passed on to the subsequent generations based on the fitness function which guides the search procedure. We use Tournament Selection [85] with $s = 5$ in all GA experiments. We also investigate the following GA settings: (1) Crossover: One-point vs. two-point with a crossover probability of $\{0.2, 0.4, 0.6, 0.8, 1\}$ and mutation probability of $\{0, 0.1\}$; (2) Mutation: 4 neighbour creation methods (Section 3.3) with a mutation probability of 0.1 and 0.2; (3) Population: Size of $\{50, 100, 200\}$ with $\{200, 100, 50\}$ generations respectively. The final setting we used to answer our RQs consists of a GA with a population size of 50, a two-point crossover with 0.6 probability, *noise vectors* with a probability of 0.2 for mutation, and no fitness re-computation of unchanged individuals.

Weighted Sum Multi-objective Algorithm (WSM) is the multi-objective version of the techniques described above (i.e., HC, TS and GA) obtained by using the Weighted Sum method, as explained in Section 3.5, with the following set of weights W for the bias component of the function denoted by Eq. (6): W= 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%.

Non-dominated Sorting Genetic Algorithm II (NSGA-II) [70] is a well-known Multi-Objective Evolutionary Algorithm based on Pareto-optimality. NSGA-II can be considered as an extension of GAs for multiple objective function optimisation, and we use it as an example to investigate the potential benefits of multi-objective algorithms over single objective approaches, such as WSM. The fitness of an individual is determined for each objective and they are ranked based on Pareto-fronts and crowding distance. In our study, NSGA-II applies the same settings as GA.

All the above search-based approaches scale based on the number of cosine-similarity comparisons for word pairs when measuring fitness (i.e., semantic correctness and bias): This is directly dependent on the size and number of training sets as well as the dimensionality of the word embeddings.

To account for randomness in stochastic search, we perform 100 independent repetitions of each experiment. Each experiment is limited to 10,000 fitness evaluations and the average results achieved across multiple runs are reported. The parameters of HC, TS, GA and NSGA-II are tuned on the MEN dataset, which is common to all experiments.

Table 2 summarises the parameter settings for each of the approaches.

Table 2
Parameter settings of the search algorithms.

Search algorithm	Parameters
Random Search	- Mutation operator: Noise vector - Level of noise: 0.05
Hill Climbing	- Mutation operator: Noise value - Level of noise: 0.14
Tabu Search	- Mutation operator: Noise value - Level of noise: 0.4 - Tabu list size: 150
GA	- Population size: 50 - Selection: Tournament selection of size 5 - Crossover probability: 0.6 - Crossover type - Two-point crossover - Mutation probability - 0.2 - Mutation operator - Noise vector - Level of nose - 0.05
NSGA-II	- Population size: 50 - Selection: Tournament selection of size 5 - Crossover probability: 1.0 - Crossover type - Two-point crossover - Mutation probability - 0.2 - Mutation operator - Noise vector - Level of nose - 0.05

4.6. Validation and evaluation criteria

In order to validate the effectiveness of search-based approaches to optimise gender bias, we use W6, W7 and W8 in turn as training set, and each of the remaining two as a test set (e.g., we train each of the search-based approaches on W6 and test their effectiveness on W7 and W8, separately). Therefore, each search method is evaluated on six different, train–test combinations.

The performance of each search method on each of the test sets is illustrated with boxplots. We also use statistical significance tests to assess differences in the performance of the algorithms. Since many of the samples come from non-normally distributed populations, we use the Wilcoxon Signed-Rank test [86], which is a non-parametric test that makes no assumptions about underlying data distributions. We set the confidence limit, α, at 0.05 and apply the standard Bonferroni correction (α/K, where K is the number of hypotheses) when multiple hypotheses are tested. In particular, depending on the RQ, we test the following null hypothesis:

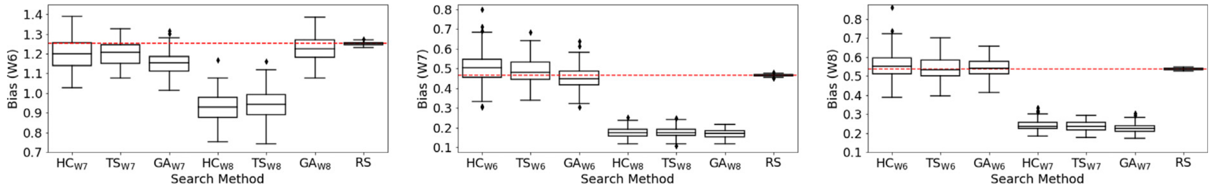
(RQ1) H_0: *The gender bias achieved by approach*$_x$ *is not lower than the one achieved by approach*$_y$. The alternative hypothesis is as follows: H_1: *The gender bias achieved by approach*$_x$ *is lower than the one achieved by approach*$_y$.

(RQ2) H_0: *The semantic correctness achieved by approach*$_x$ *is lower than the one achieved by approach*$_y$. The alternative hypothesis is as follows: H_1: *The semantic correctness achieved by approach*$_x$ *is not lower than the one achieved by approach*$_y$.

We summarise the results of the Wilcoxon tests by using the following win-tie-loss procedure [72,87,88]: We count the number of times an approach scored a p–value <0.01 (win), p–value >0.99 (loss), and 0.01≤ p–value ≥0.99 (tie).

We also used the Vargha Delaney's $\hat{A}_{12}$ standardised non-parametric effect size measure in order to verify whether any statistical significant difference is worthy of practical interest [89]. $\hat{A}_{12}$ is computed based on the following formula $\hat{A}_{12} = (R_1/m - (m+1)/2)/n$, where R_1 is the rank sum of the first data group we are comparing, and m and n are the number of observations in the first and second data sample, respectively. If the two algorithms are equivalent, then $\hat{A}_{12} = 0.5$. Respectively, an $\hat{A}_{12}$ higher than 0.5 denotes that the first algorithm is more likely to produce better results. We consider an effect size $\hat{A}_{12} \geq 0.72$ as large, $0.64 < \hat{A}_{12} < 0.72$ as medium, and $0.56 < \hat{A}_{12} \leq 0.64$ as small, although these thresholds are not definitive [71]. Since we are interested in *any* improvement, no transformation is performed on the $\hat{A}_{12}$ [90].

To answer RQ3, we use a graphical comparison of the results achieved by NSGA-II, WSM (i.e., HC_{WSM}, TS_{WSM}, GA_{WSM}), single-objective search methods and the state-of-the-art approaches HD and LP, both in terms of correctness and fairness. Indeed, we need to quantify the overall quality of prediction models with respect to both objectives at the same time, as analysing the solutions looking only at one objective at time does not give us information about the trade-off between these two competing objectives [91]. Therefore, we use scatter plots to visualise and compare the results of these methods by showing their non-dominated solutions in terms of gender bias and semantic correctness. This allows us to assess the overall quality of word embedding models based on the trade-off achieved by the two equally important



(a) W2V - Tested on W6 (b) W2V - Tested on W7 (c) W2V - Tested on W8

Fig. 1. RQ1: Boxplots of gender bias values (the lower the better) achieved by the single-objective search algorithms (HC, TS, GA) and RS over 100 runs. The dashed (red) line indicates the gender bias value of the original pre-trained model baseline.

objectives of fairness and correctness, as recommended in the literature [72,91–93].

4.7. Threats to validity

As in any empirical study, we cannot claim that our results apply to any other subject and technique but those investigated herein. However, we strove to provide a detailed explanation of our approach and the experimental design in order for other researchers to verify, replicate and extend our work. Moreover, and to further facilitate this, we make our scripts and data publicly available [33]. We investigate the effectiveness of our proposal for one of the most popular word embedding models (i.e., W2V pre-trained on Google News). Other models (e.g.,GloVe [25] and fastText [64]) as well as other training corpora (e.g., GloVe models pre-trained on Twitter data or Wikipedia articles) could have been investigated. However, as our focus lies on evaluating our adaptation approach and not the best word embedding approach, we decided to use one of the most popular models pre-trained on a supposedly objective domain such as news articles. This model is also publicly available and widely used in previous work. The search algorithms we investigated may perform differently with other settings, however, to mitigate this threat we experimented with a wide variety of settings for each approach on a common dataset to identify the best combination for each of the algorithms, so to tune each of them to their best, as detailed in Section 4.5. In order to mitigate possible bias arising from the randomness of stochastic searches, we perform 100 independent repetitions of each experiment and use both statistical significance tests and effect size to analyse the results. We carefully apply the statistical tests, verifying all the required assumptions (e.g., we applied non-parametric tests as we cannot assume a normal distribution of the data) and corrected for multiple hypotheses statistical testing, to reinforce the conclusion validity of our study.

5. Empirical study results

This section presents the results we gathered in our empirical study to answer the research questions presented in Section 4.

5.1. RQ1: Single-objective optimisation

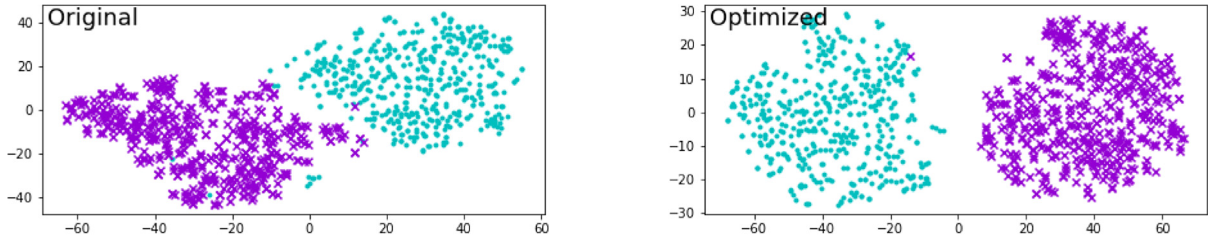
Fig. 1 shows the boxplots of the gender bias values achieved by each of the single-objective search algorithms (i.e., HC, TS, GA) and RS on the test sets over 100 runs. Each of the algorithms is denoted with the training set used to train for bias (e.g., HC_{W7} denotes that HC has been trained on W7). Fig. 1 also shows the performance of the baseline pre-trained original models W2V (i.e., Base), denoted by a dashed line. We can observe that search-based methods are able to reduce gender bias and provide better results than both RS and the pre-trained original models in 67% of the cases considered (more details in Table 3). Fig. 1 also shows consistency among the performance of the search-based approaches with respect to the test set. Specifically, all search-based approaches trained on either W7 or W8 outperform the original baseline models and RS, in all cases considered. While, when we train them on W6, GA is the only approach able to build debiased models achieving better results than the baseline and RS for W7 (Fig. 1: b). and similar ones for the W8 test set (Fig. 1: c).. This suggests that local search techniques, such as HC and TS, might overfit on W6, given that they are more prone to get stuck in local optima than global search-based approaches like GA. On the other hand, all search-based approaches outperform both benchmarks when trained on W7 and tested on W8 (Fig. 1: c) and vice versa (Fig. 1: b). Overall, we observe that a lower bias is achieved when the search-based approaches are trained and tested on more similar datasets (i.e., W7 and W8 have common attribute and target words). However, even when the training and testing data are more dissimilar (i.e., W6 and W7, W8), the search-based approaches are still able to reduce the bias present in the original pre-trained models. As shown in Fig. 1: a, when trained on each of W7 and W8, and tested on W6, all search-based models obtain lower bias values than those produced by both, Base and RS.

Table 3 summarises the results of the Wilcoxon Test comparing each of the approaches listed in the columns with those listed in the rows, for all pairs of train and test sets, in the form of win-tie-loss, as outlined in Section 4.6. It also reports the results of the Â12 statistic measure for all pair comparisons where a win is observed.

We observe that RS and Base have the poorest performance (lowest number of wins) among all methods considered and that there is no difference in their performance. On the other hand, GA achieves statistically significantly better performance than Base in 67% of the cases, and never worse in the others. HC and TS also outperform Base, with statistical significance, on 67% (of the cases each and lose in 17% of the cases. The same observation can be made when comparing the approaches with RS. The performance of HC and TS is very similar; indeed when both approaches are compared, they obtain ties in all cases. However, they tend to perform similarly (42% of the cases) or worse (42% of the cases) than GA in the majority of the cases, with only 17% (2 out of 12) of the cases being better. Overall, GA is the best performing algorithm for word embedding models achieving wins on 54% of the cases (13 out of 24) with 46% of them having large effect sizes, 23% having medium and 31% having small ones.

Fig. 2 provides an example of the bias mitigation behaviour of our approach by showing the vectors before optimisation (i.e., from the original pre-trained model), and after optimisation.[9] In this example, we use W6 as a training set, as it exhibits the highest degree of bias for W2V, and Tabu Search for optimisation, as it shows the lowest degree of bias for W6. We can observe that the male and female terms are clearly separated

[9] To this end we follow the procedure of Gonen and Goldberg [59], who used tSNE [94] to visualise 1000 word embeddings (i.e., the 500 most female-biased and 500 most male-biased terms) in two dimensions.



(a) Original.

(b) Optimised.

Fig. 2. RQ1: Visualisation of bias mitigation effectiveness with tSNE. (a) shows the most male-biased and female-biased vectors of the original, pre-trained W2V model; (b) shows vectors after optimisation obtained using Tabu Search with WEAT 6 as a training set.

Table 3
RQ1: Win-tie-loss summary of the Wilcoxon tests comparing gender bias achieved by each pair of methods (columns vs. rows) on all test sets. The last row shows the $\hat{A}_{12}$ effect size (large/med/small) of the total wins achieved by a given method.

	Base	RS	HC	TS	GA
Base	–	0-3-0	4-1-1	4-1-1	4-2-0
RS	0-3-0	–	4-1-1	4-1-1	4-2-0
HC	1-1-4	1-1-4	–	0-6-0	3-2-1
TS	1-1-4	1-1-4	0-6-0	–	2-3-1
GA	0-2-4	0-2-4	1-2-3	1-3-2	–
Total	2-7-12	2-7-12	9-10-5	9-11-4	13-9-2
A12	0/1/1	0/1/1	9/0/0	9/0/0	6/3/4

after performing the optimisation procedure with Tabu Search. A clear separation of gendered terms is in line with WEAT tests, as long as the distance to neutral words is comparable.

> Answer to RQ1: Local and global search techniques are able to reduce gender bias in word embeddings by producing, on average, models with significantly less bias than the original pre-trained ones in 67% of the cases. GA is the best performing approach as it always generates word embeddings having a statistically significantly lower or similar gender bias than those generated by the two benchmarks (i.e., Base and RS) with large effect size in 46% of the cases, and produces similar or statistically significant better results than HC and TS in 83% of the cases, while being worse in only 17%.

5.2. RQ2: Detrimental effects

Table 4 shows the mean semantic correctness values achieved by the word embedding models built by HC, TS and GA using gender bias as fitness function over 100 runs, and the semantic correctness values of the original word embedding model (i.e., Base). It also reports the p-values of the Wilcoxon test along with the Â12 statistic measure obtained when comparing the performance of each search-based approach with that of Base. We can observe that the mean semantic correctness achieved by all the search algorithms are lower than the ones achieved by Base in 89% of all cases studied (8 out of 9), highlighting a detrimental effect. The Wilcoxon Test and Â12 results also support this observation as they show that the difference in performance between Base and the search-based approaches is always statistically significant in favour of the latter with the effect size being large in all cases. Among the three search approaches (HC, TS, GA), GA has the best semantic correctness independent of the WEAT training set. Therefore, it is interesting to evaluate search approaches in a multi-objective setting which takes both semantic correctness and fairness into account during the optimisation procedure.

Table 4
RQ2. Semantic correctness achieved by the original word embeddings (Base) and the search methods HC, TS, and GA (mean values over 100 runs) using a single-objective fitness function based on gender bias. The Wilcoxon test and effect sizes results with respect to Base are also shown.

	Semantic correct.	p-value ($\hat{A}_{12}$)
Base	0.77	n.a.
HC_{W6}	0.74	0.00 (1.00)
TS_{W6}	0.74	0.00 (1.00)
GA_{W6}	0.76	0.00 (1.00)
HC_{W7}	0.75	0.00 (1.00)
TS_{W7}	0.75	0.00 (1.00)
GA_{W7}	0.77	0.00 (0.90)
HC_{W8}	0.71	0.00 (1.00)
TS_{W8}	0.71	0.00 (1.00)
GA_{W8}	0.76	0.00 (1.00)

> Answer to RQ2: Using single-objective search that minimises word embedding gender bias statistically significantly reduces their semantic correctness. This detrimental effect is observed for all single-objective search-based approaches.

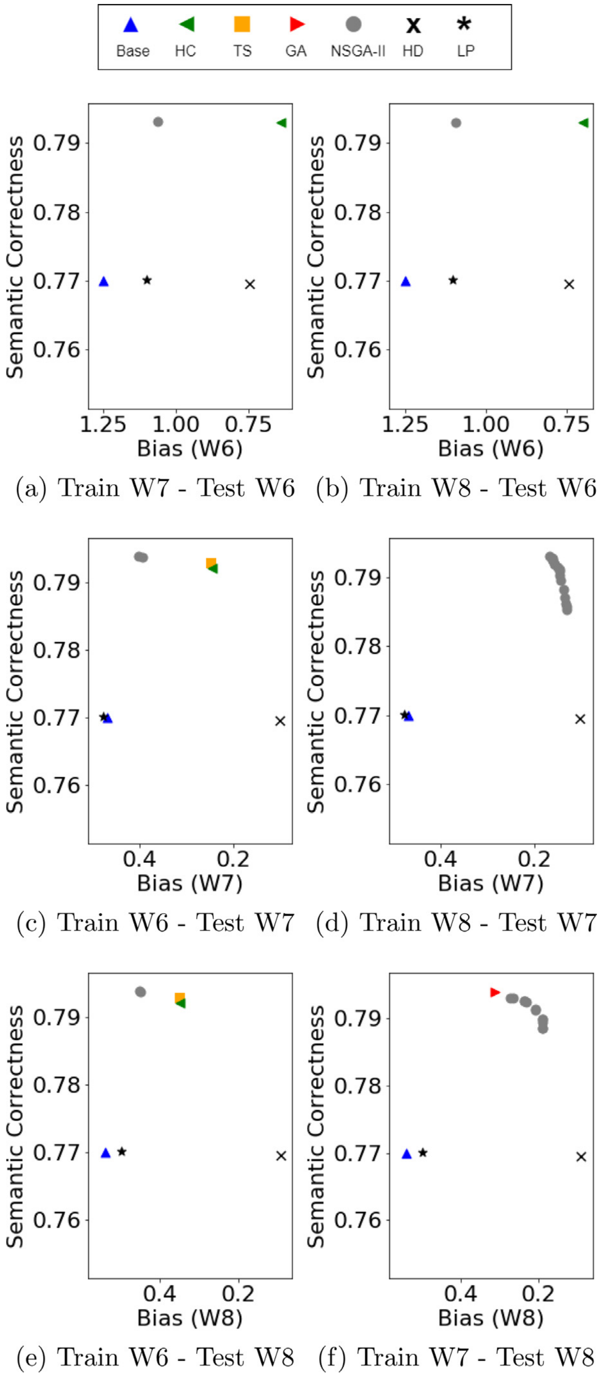
5.3. RQ3: Multi-objective optimisation

RQ3 investigates whether the use of multi-objective optimisation (either based on the WSM or Pareto-optimality approach) allows us to avoid the detrimental effect observed on semantic correctness in RQ2, and therefore to improve both semantic correctness and bias of word embedding models at the same time.

In Fig. 3, we graphically compare the trade-off between semantic correctness and bias, achieved by all non-dominated solutions obtained across all approaches considered, i.e., NSGA-II, WSM, and single-objective algorithms. Additionally, we display the results achieved by Base and the state-of-the-art HD and LD. Note that in the graphs, we invert the x-axis (*bias*) to facilitate the interpretation and analysis of the results. Therefore, in order for a solution to dominate another it should be displayed above it (meaning it achieves a better semantic correctness) and to the right of it (meaning it achieves a better fairness).[10]

As it can be seen from Fig. 3, the solutions produced by multi-objective search (either NSGA-II, or WSM for HC, TS and GA) dominate the Base solution for all cases considered, thus providing a better correctness–fairness trade-off. Moreover, none of the

[10] In the literature, quality indicators have been used to quantify the quality of trade-offs achieved by algorithms and their Pareto-front, when dealing with multiple objectives [91]. An evaluation of the algorithms with respect to such Pareto-front quality indicators can be found in our online repository [33].



(a) Train W7 - Test W6 (b) Train W8 - Test W6

(c) Train W6 - Test W7 (d) Train W8 - Test W7

(e) Train W6 - Test W8 (f) Train W7 - Test W8

Fig. 3. RQ3: Bias and semantic correctness achieved by the Base original word embedding model, NSGA-II (circle) and WSM (filled markers), and the state-of-the-art HD and LP.

single-objective approaches explored in RQs 1–2 produce Pareto-optimal solutions when compared to the solutions produced by NSGAII and WSM, i.e., the solutions produced by single-objective approaches are all outperformed (dominated) by the solutions provided by multi-objective approaches. This signifies that both Base solution and single-objective produced solutions are always dominated by at least one of the two multi-objective approaches we explored, and reveals that using multi-objective approaches is the best way to achieve an optimal trade-off between fairness and semantic correctness.

Table 5
Runtime in seconds of the search approaches for single objectives (semantic correctness, fairness) and multiple objectives (semantic correctness and fairness combined). Averages over 100 repetitions are shown.

	Search algorithm	Runtime in s	Standard deviation
Fairness	Hill Climbing	35	0.2
	Tabu Search	35	0.4
	GA	35	0.4
Semantic	Hill Climbing	264	1.1
	Tabu Search	277	4.8
	GA	284	2.7
Multiple	Hill Climbing	304	1.0
	Tabu Search	300	1.1
	GA	293	1.1
	NSGA-II	299	2.6

If we focus our analysis on determining whether there is a best performing multi-objective approach, the answer seems to be in favour of NSGA-II, at least from a purely quantitative point of view. Indeed, this algorithm is almost always able to provide a non-dominated solution for each pair of train and test set we investigated (i.e., there is an NSGA-II solution for 6 out of the 6 graphs shown in Fig. 3), even when WSM is not able to produce one (see Fig. 3: d). These result are in line with those by Chen and Li [95], who showed that Pareto search is preferred over weighted search for problems in the Search-Based Software Engineering domain.

From a more qualitative perspective, we observe that when an engineer is interested in optimising fairness while making sure accuracy does not deteriorate, their natural choice should be NSGA-II, since it is able to strictly dominate Base for every test set. On the other hand, if an engineer is willing to sacrifice semantic correctness for fairness in the design of word embedding models, then WSM might be a better choice given that in four out of the six cases (67%), they produce solutions with the highest fairness albeit at the cost of correctness.

Benchmarking our approach against the state-of-the-art, HD [20] and LP [29], reveals that our approach always generates word embeddings with a higher semantic correctness than those achieved by HD and LP, and a higher fairness (i.e., lower bias) than HD and LP in 33% and 100% of the cases, respectively. This shows that our approach offers a valuable alternative for practitioners that are interested in improving both semantic correctness and fairness.

> Answer to RQ3: Multi-objective optimisation methods are able to avoid the detrimental effect encountered when using single-objective methods (RQ2), by reducing bias and improving semantic correctness at the same time. While both, WSM and NSGA-II, always provide at least one solution which dominates the original pre-trained word embedding models, NSGA-II might be preferable when the semantic correctness of the models needs to be guaranteed, while WSM is preferable when one wants to optimise word embeddings for fairness albeit sacrificing some correctness.

5.4. Complexity analysis

In addition to investigating the effectiveness of our search approaches to achieve improvement in semantic correctness and fairness, we investigated their runtime. This is important to assess whether improving existing word embeddings is more efficient than training a new model from scratch.

Table 5 shows the runtime of each search approach, averaged over 100 repetitions. While no exact measurements are provided on how much effort was required to train the W2V model on news articles, a blog post states that it required "about 9 h on multi-core machine", without further details on processing power.[11] In contrast, our approaches, given the limit of 10,000 fitness evaluations, require 5 min or less when optimising for semantic correctness, and only 35 s when optimising for fairness. Our experiments were performed on a SGE v8.1.9 grid system with nodes using Intel Skylake CPUs with 3.5 GHz frequency and up to 3TB of RAM, whereas our experiments require less than 4 GB.

The runtime of the search procedure is not dependent on the pre-trained word embedding size, but only on the size of the training sets (i.e., the number of cosine-similarity comparisons performed). The MEN training set performs 2,000 comparisons; each of the three WEAT test sets performs 128 comparisons.

The space complexity of the search procedure is solely depending on the number of unique words considered in the training sets, and overhead incurred by using the different search algorithms. Therefore, our optimisation is independent of the size of the pre-trained word embedding model under investigation.

6. Conclusions and future work

In this paper, we have investigated the effectiveness of local and global search techniques to optimise pre-trained word embedding models for a single objective (i.e., gender bias) and multiple objectives (i.e., semantic correctness and gender bias simultaneously). We found that single-objective search techniques (local and global) can be used to reduce gender bias of word embedding models. Among those, GA is the best performing technique overall, while HC and TS tend to perform similarly to one another. However, our findings also show that optimising bias solely comes at the cost of sacrificing semantic correctness. In fact, we observe that all search-based approaches achieve lower semantic correctness than the baselines (i.e., Base and RS) when guided by a single-objective function optimising bias. This prompts the need for approaches that can optimise both, bias and semantic correctness, at the same time. We therefore investigate the application of such multi-objective search-based approaches (either based on the weighted sum approach or on Pareto-optimality). Our results show that multi-objective search is able to reduce gender bias and at the same time improve their semantic correctness. We also observe that NSGA-II is able to provide a non-dominated solution for all six test sets, therefore providing word embeddings with a higher semantic correctness and fairness than the Base model. If fairness improvements are prioritised, WSM are preferable, as they achieve a higher fairness than NSGA-II in four out of six cases albeit at the cost of correctness. Additionally, our approach is able to create word embeddings of higher semantic correctness than two state-of-the-art techniques in all cases, while also achieving a higher degree of fairness in 67% of the cases.

Our proposed multi-objective approach enables the engineers to explore the trade-offs between two important competing objectives (accuracy and fairness) among a rich set of equally viable solutions to the problem at hand, while previous work only offered the engineer a single proposed solution, which is not realistic when they face problems constituted of competing goals [72, 96].

This opens up a rich agenda of future work. In addition to investigating other pre-trained word embedding models and semantic evaluation measures, future work can investigate the effectiveness of our proposal to tackle additional fairness aspects, such as race and age. This can be achieved by investigating multi-objective algorithms as well as many-objective ones to optimise more than one fairness aspect at the same time. Besides, our proposed approach can be easily extended to work with specific language models. In the context of software systems, a fairness metric that is able to access the usage of the language model within the software (e.g., for search queries) can be derived. Given this metric, our multi-objective approach can then be used to optimise both, fairness and the performance of the software system.

Moreover, as our approach focuses on measuring intrinsic biases of word embedding models (i.e., bias residing in the embedding vectors) [97,98], an interesting avenue of future work are extrinsic (i.e., downstream) tasks to determine the fairness of word embedding models (e.g., co-reference resolution or hate speech detection).

CRediT authorship contribution statement

Max Hort: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Rebecca Moussa:** Methodology, Validation, Writing – review & editing. **Federica Sarro:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

M. Hort, R. Moussa and F. Sarro are supported by the European Research Council (ERC) Grant No. 741278 (EPIC).

References

[1] S.A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E.P. Hamilton, D. Roth, A comparative study of fairness-enhancing interventions in machine learning, in: Procs. of the Conference on Fairness, Accountability, and Transparency, ACM, 2019, pp. 329–338.

[2] J. Horkoff, Non-functional requirements for machine learning: Challenges and new directions, in: 2019 IEEE 27th International Requirements Engineering Conference, RE, IEEE, 2019, pp. 386–391.

[3] J.M. Zhang, M. Harman, L. Ma, Y. Liu, Machine learning testing: Survey, landscapes and horizons, TSE (2020) 1.

[4] J. Zhang, M. Harman, "Ignorance and prejudice" in software fairness, in: 2021 IEEE/ACM 43th International Conference on Software Engineering, ICSE.

[5] Z. Chen, J.M. Zhang, M. Hort, F. Sarro, M. Harman, Fairness testing: A comprehensive survey and analysis of trends, 2022, arXiv preprint arXiv:2207.10223.

[6] M. Hort, Z. Chen, J.M. Zhang, F. Sarro, M. Harman, Bias mitigation for machine learning classifiers: A comprehensive survey, 2022, arXiv preprint arXiv:2207.07068.

[7] J. Angwin, J. Larson, S. Mattu, L. Kirchner, Machine bias. ProPublica, 2016, See https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[8] R. Berk, H. Heidari, S. Jabbari, M. Kearns, A. Roth, Fairness in criminal justice risk assessments: The state of the art, Sociol. Methods Res. (2018) 0049124118782533.

[9] A. Mukerjee, R. Biswas, K. Deb, A.P. Mathur, Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management, Int. Trans. Oper. Res. 9 (5) (2002) 583–597.

[10] J. Zhao, Y. Zhou, Z. Li, W. Wang, K.-W. Chang, Learning gender-neutral word embeddings, EMNLP (2018) 4847–4853.

[11] https://groups.google.com/g/word2vec-toolkit/c/lxbl_MB29Ic/m/NDLGId3KPNEJ

[11] J. Dastin, Amazon scraps secret AI recruiting tool that showed bias against women, 2018, URL https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G.

[12] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Procs. of the 10th ACM Conference on Recommender Systems, 2016, pp. 233–240.

[13] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Procs. of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 425–434.

[14] Y. Ren, D. Ji, Neural networks for deceptive opinion spam detection: An empirical study, Inform. Sci. 385 (2017) 213–224.

[15] T. Kenter, M. De Rijke, Short text similarity with word embeddings, in: Procs. of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1411–1420.

[16] E. Nalisnick, B. Mitra, N. Craswell, R. Caruana, Improving document ranking with dual word embeddings, in: Procs. of the 25th International Conference Companion on World Wide Web, 2016, pp. 83–84.

[17] F. Calefato, F. Lanubile, F. Maiorano, N. Novielli, Sentiment polarity detection for software development, Empir. Softw. Eng. 23 (3) (2018) 1352–1382.

[18] X. Yang, D. Lo, X. Xia, L. Bao, J. Sun, Combining word embedding with information retrieval to recommend similar bug reports, in: Procs. of the International Symposium on Software Reliability Engineering, ISSRE, IEEE, 2016, pp. 127–137.

[19] X. Ye, H. Shen, X. Ma, R. Bunescu, C. Liu, From word embeddings to document similarities for improved information retrieval in software engineering, in: Procs. of the International Conference on Software Engineering, ICSE, 2016, pp. 404–415.

[20] T. Bolukbasi, K.-W. Chang, J.Y. Zou, V. Saligrama, A.T. Kalai, Man is to computer programmer as woman is to homemaker? Debiasing word embeddings, in: Advances in Neural Information Processing Systems, 2016, pp. 4349–4357.

[21] A. Caliskan, J.J. Bryson, A. Narayanan, Semantics derived automatically from language corpora contain human-like biases, Science 356 (6334) (2017) 183–186.

[22] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, K.-W. Chang, Men also like shopping: Reducing gender bias amplification using corpus-level constraints, in: Procs. of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2979–2989.

[23] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, K.-W. Chang, Gender bias in coreference resolution: Evaluation and debiasing methods, in: Procs. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 15–20, http://dx.doi.org/10.18653/v1/N18-2003, URL https://www.aclweb.org/anthology/N18-2003.

[24] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[25] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Procs. of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.

[26] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, A. Roth, A convex framework for fair regression, in: FAT-ML Workshop, 2017.

[27] T. Kamishima, S. Akaho, H. Asoh, J. Sakuma, Fairness-aware classifier with prejudice remover regularizer, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 35–50.

[28] J. Zhao, T. Wang, M. Yatskar, R. Cotterell, V. Ordonez, K.-W. Chang, Gender bias in contextualized word embeddings, in: Procs. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 629–634, http://dx.doi.org/10.18653/v1/N19-1064, URL https://www.aclweb.org/anthology/N19-1064.

[29] S. Dev, J. Phillips, Attenuating bias in word vectors, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 879–887.

[30] S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, Y. Goldberg, Null it out: Guarding protected attributes by iterative nullspace projection, 2020, arXiv preprint arXiv:2004.07667.

[31] A. Lauscher, G. Glavaš, S.P. Ponzetto, I. Vulić, A general framework for implicit and explicit debiasing of distributional word vector spaces, in: Procs. of the AAAI Conference on Artificial Intelligence, vol. 34, (no. 05) 2020, pp. 8131–8138.

[32] E. Bruni, N.-K. Tran, M. Baroni, Multimodal distributional semantics, J. Artificial Intelligence Res. 49 (2014) 1–47.

[33] On-line appendix, URL https://figshare.com/s/99e80ca8b59c19635b44.

[34] Y. Brun, A. Meliou, Software fairness, in: Procs. of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2018, pp. 754–759.

[35] F. Kamiran, T. Calders, Data preprocessing techniques for classification without discrimination, Knowl. Inf. Syst. 33 (1) (2012) 1–33.

[36] M. Feldman, S.A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and removing disparate impact, in: Procs. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 259–268.

[37] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, A. Huq, Algorithmic decision making and the cost of fairness, in: Procs. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 797–806.

[38] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, 2019, arXiv preprint arXiv:1908.09635.

[39] D. Pessach, E. Shmueli, Algorithmic fairness, 2020, arXiv preprint arXiv:2001.09784.

[40] S. Caton, C. Haas, Fairness in machine learning: A survey, 2020, arXiv preprint arXiv:2010.04053.

[41] J. Chakraborty, S. Majumder, Z. Yu, T. Menzies, Fairway: A way to build fair ML software, in: Procs. of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020, pp. 654–665.

[42] A. Finkelstein, M. Harman, S.A. Mansouri, J. Ren, Y. Zhang, A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making, Requir. Eng. 14 (4) (2009) 231–245.

[43] M. Hort, J. Zhang, F. Sarro, M. Harman, Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods, in: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021.

[44] S. Biswas, H. Rajan, Do the machine learning models on a crowd sourced platform exhibit bias? An empirical study on model fairness, in: Procs. of ESEC/FSE, 2020, pp. 642–653.

[45] Z. Chen, J. Zhang, F. Sarro, M. Harman, MAAT: A novel ensemble approach to addressing fairness and performance bugs for machine learning software, in: The ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE, 2022.

[46] M. Hort, F. Sarro, Did you do your homework? Raising awareness on software fairness and discrimination, in: 2021 36th IEEE/ACM International Conference on Automated Software Engineering, ASE.

[47] N. Mrkšić, D.O. Séaghdha, B. Thomson, M. Gašić, L. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, S. Young, Counter-fitting word vectors to linguistic constraints, 2016, arXiv preprint arXiv:1603.00892.

[48] I. Vulić, N. Mrkšić, Specialising word vectors for lexical entailment, 2017, arXiv preprint arXiv:1710.06371.

[49] P. Kolyvakis, A. Kalousis, D. Kiritsis, Deepalignment: Unsupervised ontology matching with refined word vectors, in: Procs. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 787–798.

[50] J.-K. Kim, M.-C. de Marneffe, E. Fosler-Lussier, Adjusting word embeddings with semantic intensity orders, in: Procs. of the 1st Workshop on Representation Learning for NLP, 2016, pp. 62–69.

[51] C. Yonghe, H. Lin, L. Yang, Y. Diao, S. Zhang, F. Xiaochao, Refining word reesprentations by manifold learning, in: Procs. of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 5394–5400.

[52] L.-C. Yu, J. Wang, K.R. Lai, X. Zhang, Refining word embeddings using intensity scores for sentiment analysis, IEEE/ACM Trans. Audio, Speech, Lang. Process. 26 (3) (2017) 671–681.

[53] M. Faruqui, J. Dodge, S.K. Jauhar, C. Dyer, E. Hovy, N.A. Smith, Retrofitting word vectors to semantic lexicons, 2014, arXiv preprint arXiv:1411.4166.

[54] K. Patel, D. Patel, M. Golakiya, P. Bhattacharyya, N. Birari, Adapting pre-trained word embeddings for use in medical coding, in: BioNLP 2017, 2017, pp. 302–306.

[55] M. Hort, F. Sarro, Optimising word embeddings with search-based approaches, in: Procs. of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 269–270.

[56] Y. Luo, J. Tang, J. Yan, C. Xu, Z. Chen, Pre-trained multi-view word embedding using two-side neural network, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

[57] P. Kameswara Sarma, Y. Liang, B. Sethares, Domain adapted word embeddings for improved sentiment classification, in: Procs. of the Workshop on Deep Learning Approaches for Low-Resource, NLP, Association for Computational Linguistics, Melbourne, 2018, pp. 51–59, http://dx.doi.org/10.18653/v1/W18-3407, URL https://www.aclweb.org/anthology/W18-3407.

[58] T. Sun, A. Gaut, S. Tang, Y. Huang, M. ElSherief, J. Zhao, D. Mirza, E. Belding, K.-W. Chang, W.Y. Wang, Mitigating gender bias in natural language processing: Literature review, 2019, arXiv preprint arXiv:1906.08976.

[59] H. Gonen, Y. Goldberg, Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them, 2019, arXiv preprint arXiv:1903.03862.

[60] M. Kaneko, D. Bollegala, Gender-preserving debiasing for pre-trained word embeddings, 2019, arXiv preprint arXiv:1906.00742.

[61] S. Shin, K. Song, J. Jang, H. Kim, W. Joo, I.-C. Moon, Neutralizing gender bias in word embedding with latent disentanglement and counterfactual generation, 2020, arXiv preprint arXiv:2004.03133.

[62] M. Kaneko, D. Bollegala, Dictionary-based debiasing of pre-trained word embeddings, in: Proc. of the 16th European Chapter of the Association for Computational Linguistics, EACL, 2021.

[63] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Procs. of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 746–751.

[64] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2016) http://dx.doi.org/10.1162/tacl_a_00051.

[65] M. Faruqui, Y. Tsvetkov, P. Rastogi, C. Dyer, Problems with evaluation of word embeddings using word similarity tasks, 2016, arXiv preprint arXiv:1605.02276.

[66] C. Spearman, The proof and measurement of association between two things, Am. J. Psychol. 15 (1904) 72–101.

[67] M. Faruqui, C. Dyer, Community evaluation and exchange of word vectors at wordvectors. org, in: Procs. of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 19–24.

[68] Y. Tsvetkov, M. Faruqui, W. Ling, G. Lample, C. Dyer, Evaluation of word vector representations by subspace alignment, in: Procs. of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2049–2054.

[69] M. Harman, P. McMinn, J.T. De Souza, S. Yoo, Search based software engineering: Techniques, taxonomy, tutorial, in: Empirical Software Engineering and Verification, Springer, 2010, pp. 1–59.

[70] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[71] F. Sarro, A. Petrozziello, M. Harman, Multi-objective software effort estimation, in: Procs. of the International Conference on Software Engineering, ICSE, IEEE, 2016, pp. 619–630.

[72] F. Sarro, F. Ferrucci, M. Harman, A. Manna, J. Ren, Adaptive multi-objective evolutionary algorithms for overtime planning in software projects, IEEE TSE 43 (10) (2017) 898–917.

[73] P. Ralph, S. Baltes, D. Bianculli, Y. Dittrich, M. Felderer, R. Feldt, A. Filieri, C.A. Furia, D. Graziotin, P. He, et al., Proposed ACM SIGSOFT standard for optimization studies in SE (including SBSE), 2020, URL https://github.com/Greg4cr/sbse-sigsoft-standard.

[74] P. Ralph, S. Baltes, D. Bianculli, Y. Dittrich, M. Felderer, R. Feldt, A. Filieri, C.A. Furia, D. Graziotin, P. He, et al., ACM SIGSOFT empirical standards, 2020, arXiv preprint arXiv:2010.03525.

[75] L.K. Şenel, İ. Utlu, V. Yücesoy, A. Koç, T. Çukur, Semantic structure and interpretability of word embeddings, IEEE/ACM Trans. Audio, Speech, Lang. Process. 26 (10) (2018) 1769–1779.

[76] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Genetic programming for effort estimation: An analysis of the impact of different fitness functions, in: 2nd International Symposium on Search Based Software Engineering, 2010, pp. 89–98.

[77] B.A. Nosek, M.R. Banaji, A.G. Greenwald, Harvesting implicit group attitudes and beliefs from a demonstration web site, Group Dyn.: Theory, Res. Pract. 6 (1) (2002) 101.

[78] B.A. Nosek, M.R. Banaji, A.G. Greenwald, Math=male, me=female, therefore math≠ me, J. Personal. Soc. Psychol. 83 (1) (2002) 44.

[79] A. Bakarov, A survey of word embeddings evaluation methods, 2018, arXiv preprint arXiv:1801.09536.

[80] T. Schnabel, I. Labutov, D. Mimno, T. Joachims, Evaluation methods for unsupervised word embeddings, in: Procs. of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 298–307.

[81] P. Ralph, N.b. Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri, et al., Empirical standards for software engineering research, 2020, arXiv preprint arXiv:2010.03525.

[82] F. Glover, Future paths for integer programming and links to artificial intelligence, Comput. Oper. Res. 13 (5) (1986) 533–549.

[83] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, 1988.

[84] M. Srinivas, L.M. Patnaik, Genetic algorithms: A survey, Computer 27 (6) (1994) 17–26.

[85] D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in: Foundations of Genetic Algorithms, vol. 1, Elsevier, 1991, pp. 69–93.

[86] F. Wilcoxon, Individual comparisons by ranking methods, in: Breakthroughs in Statistics, Springer, 1992, pp. 196–202.

[87] E. Kocaguneli, T. Menzies, J.W. Keung, On the value of ensemble effort estimation, IEEE TSE 38 (6) (2011) 1403–1416.

[88] F. Sarro, M. Harman, Y. Jia, Y. Zhang, Customer rating reactions can be predicted purely using app features, in: Procs. of RE, IEEE, 2018, pp. 76–87.

[89] A. Arcuri, L. Briand, A Hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering, Softw. Test. Verif. Reliab. 24 (3) (2014) 219–250.

[90] G. Neumann, M. Harman, S. Poulding, Transformed vargha-delaney effect size, in: International Symposium on Search Based Software Engineering, Springer, 2015, pp. 318–324.

[91] V. Tawosi, F. Sarro, A. Petrozziello, M. Harman, Multi-objective software effort estimation: A replication study, IEEE Trans. Softw. Eng. (2021) 1, http://dx.doi.org/10.1109/TSE.2021.3083360.

[92] G. Guizzo, F. Sarro, J. Krinke, S.R. Vergilio, Sentinel: A hyper-heuristic for the generation of mutant reduction strategies, IEEE Trans. Softw. Eng. (2020).

[93] P. Ralph, N. bin Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri, B.B.N. de França, C.A. Furia, G. Gay, N. Gold, D. Graziotin, P. He, R. Hoda, N. Juristo, B. Kitchenham, V. Lenarduzzi, J. Martínez, J. Melegati, D. Mendez, T. Menzies, J. Molleri, D. Pfahl, R. Robbes, D. Russo, N. Saarimäki, F. Sarro, D. Taibi, J. Siegmund, D. Spinellis, M. Staron, K. Stol, M.-A. Storey, D. Taibi, D. Tamburri, M. Torchiano, C. Treude, B. Turhan, X. Wang, S. Vegas, Empirical standards for software engineering research, 2021, arXiv:2010.03525.

[94] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).

[95] T. Chen, M. Li, The weights can be harmful: Pareto search versus weighted search in multi-objective search-based software engineering, ACM Trans. Software Eng. Methodol. (2022).

[96] A.S. Sayyad, T. Menzies, H. Ammar, On the value of user preferences in search-based software engineering: A case study in software product lines, in: 2013 35th International Conference on Software Engineering, ICSE, 2013, pp. 492–501, http://dx.doi.org/10.1109/ICSE.2013.6606595.

[97] E. Sesari, M. Hort, F. Sarro, An empirical study on the fairness of pre-trained word embeddings, in: Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing, 2022.

[98] S. Goldfarb-Tarrant, R. Marchant, R.M. Sánchez, M. Pandya, A. Lopez, Intrinsic bias metrics do not correlate with application bias, 2020, arXiv preprint arXiv:2012.15859.