Optimization of Bias Mitigation in Word Embeddings: a Methodological Approach

María José Zambrano*[†‡], Felipe Bravo-Marquez*[†‡]
*Department of Computer Science, University of Chile, Santiago, Chile
[†]National Center for Artificial Intelligence (CENIA), Santiago, Chile
[‡]Millennium Institute for Foundational Research on Data (IMFD), Santiago, Chile

***Abstract*—Word embeddings (WEs) often reflect biases present in their training data, and various bias mitigation and evaluation techniques have been proposed to address this. Existing benchmarks for comparing different debiasing methods overlook two factors: the choice of training words and model hyper-parameters. We propose a robust comparison methodology that incorporates them using nested cross-validation, hyper-parameter optimization, and the corrected paired Student's t-test. Our results show that when using our evaluation approach many recent debiasing methods do not offer statistically significant improvements over the original hard debiasing model.**

***Index Terms*—Bias, Word Embeddings, Natural Language Processing**

I. Introduction

Word embeddings, which encode word meanings into dense vectors based on the distributional hypothesis, have been found to exhibit stereotypical biases that affect identity groups (e.g., gender, race, religion) present in the textual data on which they are trained [1]. An example of biases found in word embeddings includes relationships such as "man is to doctor as woman is to nurse."

Previous research has addressed this issue by introducing metrics designed to quantify bias within word embedding models, such as the Word Embedding Association Test (WEAT) [2] and Relative Negative Sentiment Bias (RNSB) [3]. Additionally, debiasing algorithms have been developed to mitigate these biases by manipulating vector representations through algebraic operations. Examples include Hard Debias (HD) [1] and Double Hard Debias (DHD) [4].

A previous study [5] highlighted that existing debiasing methods cannot be readily compared using bias metrics due to three confounding factors: (1) the reliance on different word sets when applying bias mitigation algorithms, (2) leakage between the training words used by mitigation methods and the evaluation words used by metrics, and (3) inconsistencies in normalization transformations among mitigation algorithms. To address these issues, the authors developed a methodology for comparing bias mitigation algorithms that controls for these three factors.

While this methodology aims to ensure fair comparability by minimizing variability not attributable to the algorithms themselves, it overlooks two crucial sources of variability: (1) the selection of words used for training and evaluating the debiasing models, and (2) the hyper-parameters of the debiasing models. We argue that to achieve a truly fair comparison between bias mitigation algorithms, these sources of variability must also be addressed. This involves robustly controlling and comparing algorithms to ensure that the results are not influenced by specific combinations of words and hyper-parameters, thereby preventing circumstantial advantages.

In this paper, we aim to improve the fairness of algorithm comparisons by addressing the limitations of previous methodologies. Recognizing the operational similarities between bias mitigation and supervised learning algorithms, we adapt established supervised learning techniques to the bias mitigation context. Specifically, we propose a robust comparison framework that utilizes nested cross-validation [6] for hyper-parameter optimization and applies the corrected resampled t-test [7] for statistical significance testing, drawing from well-established practices in supervised learning.

II. Related Work

A. Bias Measurement

In order to quantify the bias present in word embedding models, several bias measurement metrics have been developed. Below we describe the metrics used in this study.

The Word Embedding Association Test (WEAT) [2] calculates the degree of association between a set of words representing social groups and a second set of words that, in a fair representation, should be neutral but are often biased toward one group. Additionally, the authors of WEAT proposed another metric, the WEAT Effect Size (WEAT ES), which offers a normalized measure of the difference in the distributions of associations between the biased group and the neutral word set.

Relative Norm Distance (RND) [8] measures the relative strength of association between two sets of words, as described earlier.

Relative Negative Sentiment Bias (RNSB) [3] assesses the negativity of the words associated with a social identity group,

We acknowledge the support of ANID - Millennium Science Initiative Program - Code ICN17_002 (IMFD) and the National Center for Artificial Intelligence CENIA FB210017, Basal ANID. Mara José was founded by the National Agency for Research and Development (ANID) / Scholarship Program / Magster Nacional / 2023 - 22230745 and by the Fund to Support Research, Development and Innovation with a gender perspective in STEM of the Diversity and Gender Department of the Faculty of Physical and Mathematical Sciences of the University of Chile.

based on the principle that in fair conditions, all groups should exhibit equal levels of negativity.

The Relational Inner Product Association (RIPA) [9] calculates the dot product between words in both sets, similar to previous metrics.

Finally, the Embedding Coherence Test (ECT) [10] is designed to measure the association between the two sets of words, similar to the approach used in the previous metrics.

Despite their differing methodologies for quantifying bias in word embedding models, these metrics have been integrated into a common framework within the WEFE Library [11], which also standardizes the word sets used.

Within the library, these metrics share a common interface where each metric operates over a query that receives two word sets, labeled as "target" and "attribute." The metrics then estimate the bias present in a word embedding model based on the relationships between these word sets.

As described by Badilla et al. [11], the word sets used by these metrics are:

1) **Attributes:** words that represent attitudes, traits, characteristics, occupations, among others. In a fair setting, these attributes should have equal associations with individuals from each social group (e.g. home, wedding, nurse, professional, etc).
2) **Target:** words are used to denote specific social identity groups defined by criteria such as gender, religion, or race (e.g. woman, man, daughter, son, etc).

B. Bias Mitigation

Bias mitigation algorithms for word embeddings aim to reduce the bias in the models by performing various algebraic operations on the vectors. The algorithms considered in our study are described below.

Hard Debias (HD) [1] functions by learning the "bias direction" in the embedding model and subtracting it from words that should not be biased.

Double Hard Debias (DHD) [4] is a modification of the previous method, where the authors argue that word frequency in the training corpus affects bias in the resulting word embedding model. Their algorithm aims to account for this frequency in the debiasing process.

Repulsion Attraction Neutralization (RAN) [12] is a method that aims to mitigate bias by adjusting vectors: it repels highly biased words, attracts them to their original representations, and neutralizes them relative to the bias direction.

Lastly, Half Sibling Regression (HSR) [13] treats bias as noise and employs a confounding-noise-elimination approach to remove bias from the word embedding model.

Despite their differences, these algorithms can be integrated into a common framework, as demonstrated by the WEFE library [11], which allows them to be used interchangeably. These algorithms operate by first learning the bias present in a word embedding model from a set of words, referred to as the "bias definition", and then removing this bias from a set of "objective" words.

This process is analogous to supervised learning, where a model learns from labeled data and then applies the learned transformation to new data. Moreover, these models contain hyper-parameters that must be manually set before training. Following this approach, the algorithms in the WEFE library are implemented using scikit-learn's fit-transform interface, similar to how supervised learning methods are structured.

The word sets used for the mitigation process are named differently across various algorithms but can be generalized into three categories, as proposed in [5]:

1) **Objective:** Words to which the bias mitigation process is applied. (e.g. engineer, doctor, family, etc)
2) **Bias Definition:** Word pairs derived from two contrasting identity groups utilized by mitigation algorithms to learn and address the intended bias direction. (e.g (she/him,woman/man,mom/dad, etc)
3) **Gender Specific:** words that are associated with gender by definition but do not necessarily define the identity group. (e.g. testosterone, womb, beard, etc)

C. Bias Mitigation Comparison

In [5], the authors identify three key issues affecting fair comparisons of word embeddings bias mitigation algorithms: 1) inconsistent word sets used for training, 2) data leakage between training and evaluation words, leading to inflated results, and 3) inconsistencies in normalization practices on the embedding space across different algorithms.

To address these issues and reduce variability, the authors proposed a methodology for comparing bias mitigation algorithms in a more controlled setting. Their approach involves three key steps:

1) **Standardization**: Standardizing the word sets used by both bias mitigation algorithms and bias measurement metrics to ensure consistency across comparisons.
2) **Separation of Data**: Constraining overlap of word sets between measurement metrics and mitigation algorithms, similar to train-test splitting in machine learning.
3) **Normalization Control**: Controlling vector normalization as a pre-processing step to ensure uniformity across algorithms.

This approach aims to enable fairer comparisons between algorithms by controlling for variables that might introduce variability into the results, ensuring that observed differences are attributable to the algorithms themselves rather than external factors.

Using this methodology, the authors conducted experiments comparing the algorithms under controlled conditions. The results demonstrated that when the methodology's variables are controlled, algorithm performance becomes more consistent. This consistency is measured by computing the standard deviation of metric variations for each method (HD, DHD, HSR, and RAN) before (baseline setting) and after employing the methodology. The average standard deviation across metrics (WEAT, WEAT ES, RND, RNSB, RIPA, and ECT) was also computed, and a p-value was obtained from a two-sided, twosample t-test comparing the average standard deviations between the proposed methodology and the baseline. In addition, an analysis of how each component of the methodology affects the comparison of the algorithms was performed. The results showed that while each component individually improved the consistency of the comparisons, it was the combination of all three that resulted in a statistically significant reduction in variability.

While this methodology makes a valuable contribution to comparing word embedding debiasing methods, it lacks a robust estimate of the generalization of bias reduction for each method. This limitation arises from not addressing two key sources of variability: (1) the word sets used to train and evaluate bias, and (2) the fixed hyper-parameters in the algorithms.

Neglecting these factors adds variability to the comparisons, potentially favoring one algorithm over another and compromising fairness. This underscores the need for a more comprehensive approach that accounts for these variables.

III. Proposed Methodology

In this section, we propose a methodology designed to enable a more robust comparison of bias mitigation algorithms than that of [5]. Our approach begins with hyper-parameter optimization for word embeddings debiasing algorithms using nested cross-validation. We then apply statistical testing to evaluate the performance of each algorithm, utilizing a corrected paired t-test [7]. This test is specifically designed to compare cross-validated results while accounting for the violation of independence assumptions inherent in cross-validation. This methodology is inspired by the standard approach used for statistically comparing supervised learning models and leverages the operational similarities between bias mitigation algorithms and supervised learning models.

Bias mitigation algorithms and supervised learning algorithms share operational similarities that facilitate comparative analysis. The processes of bias mitigation and measurement closely parallel the processes of training and evaluation in supervised learning models, as shown in Table I.

In supervised learning, training and evaluation typically involve using a portion of the data (the training set) to train the model, while the remaining data (the test set) is used for evaluation. A similar approach is taken in bias mitigation algorithms, where the algorithms learn and mitigate bias using a set of bias-defining words and then measure bias using a target set. Since these sets are similarly defined, they can be compared to the training and test sets, as noted in [5].

In addition to the comparisons mentioned above, another similarity between the two types of algorithms is the presence of hyper-parameters. Hyper-parameters are parameters in learning algorithms that must be set before training the classification model [14]. These parameters influence various aspects of the learning process, significantly impacting the behavior and effectiveness of the algorithm. Examples of hyper-parameters include the maximum depth in decision trees, the learning rate in gradient descent, and the number of neighbors in KNN classification.

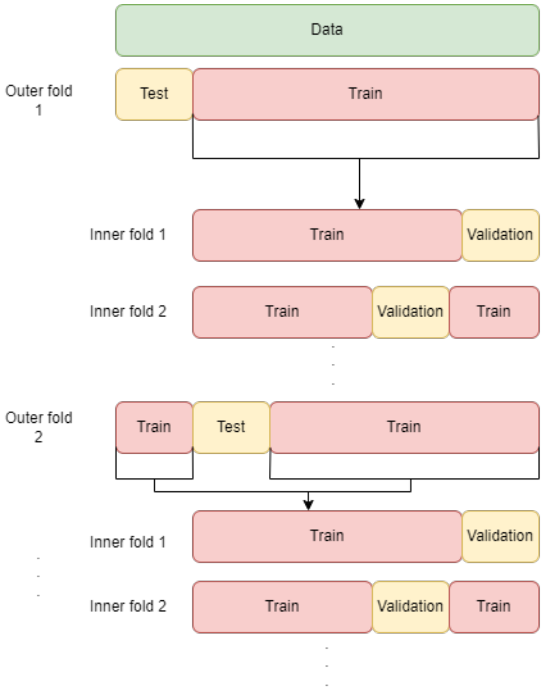

Fig. 1: Nested Cross Validation diagram.

Given the significant impact of hyper-parameters on both the learning process and the algorithmic performance, the selection of appropriate values is crucial. The goal is to identify the combination of hyper-parameters that minimizes the model error. Different techniques are used to achieve this. In our methodology, we use nested cross-validation combined with grid search, which is specifically tailored for bias reduction algorithms. This approach provides a robust assessment of generalization and helps mitigate overfitting, both critical factors when comparing bias mitigation strategies.

A. Nested Cross-Validation

Nested cross-validation [6] involves optimizing hyper-parameters by testing all possible combinations on the entire data set using a grid search. The process begins by dividing the data into outer folds, each consisting of a training and a test set. Within each iteration, the training set is further split into inner training and testing folds. A model is trained on each inner training set for every hyper-parameter combination and tested on the corresponding inner test set. The best-performing hyper-parameter combination is then used to train a model on the outer training set, which is subsequently tested on the outer test set. This process is illustrated in Figure 1.

This technique enables robust model comparison by preventing overfitting and allowing for the generalization of models to new, unseen data [15].

Aspect	**Supervised Learning**	**Bias Mitigation**
Objective	Train a model to predict an output	Reduce bias in a word embedding model
Data used	Set of labeled data	Set of word pairs
Training data	Train set, part of the set of labeled data.	Bias definition
Testing data	Test set, part of the labeled data not used for training	Target
Performance Evaluation	Testing model's performance with metrics (accuracy, precision, etc)	Measure bias with bias measurement metrics (WEAT, RND, etc)

TABLE I: Comparative table between supervised learning and bias mitigation process.

Algorithm	Hyper-parameters
HD	$svd\_solvers$
DHD	$svd\_solvers$
HSR	Alpha
RAN	Epochs, Theta, Neighbours, Learning Rate, Weights

TABLE II: Hyper-parameters in bias mitigation algorithms

B. Hyper-parameters in bias mitigation algorithms

As discussed earlier, bias mitigation algorithms, like supervised learning algorithms, also have hyper-parameters. Table II presents the hyper-parameters identified for each bias mitigation algorithm that could potentially impact their effectiveness in reducing bias. Below, we briefly describe these hyper-parameters.

1) Hard Debias: For the Hard Debias (HD) algorithm, we identify one significant hyper-parameter: *"PCA args"*, which includes the hyper-parameters used in Principal Component Analysis (PCA) [16] to derive the bias subspace. A particularly notable parameter within this category is *"$svd\_solvers$"*, which determines the solver method employed by the PCA.

2) Double Hard Debias: Similar to Hard Debias, the only hyper-parameter identified for this algorithm is the solver used in Principal Component Analysis (PCA).

3) Half Sibling Regression: In the case of Half Sibling Regression (HSR), the parameter α plays a crucial role in balancing the emphasis between the residual sum of squares and the sum of squares in the regression process. Specifically, when $\alpha = 0$, the regression reduces to a simple linear regression. According to the authors who introduced the HSR method [13], an optimal value of α is suggested to be 60.

4) Repulsion Attraction Neutralization: For Repulsion Attraction Neutralisation (RAN) we identify a total of 6 hyper-parameters used in the algorithm. These are listed below:

1) *PCA args*: Same as HD and DHD, RAN performs a PCA to find the bias subspace, meaning it includes the hyper-parametes of the PCA, where the most relevant is *"$svd\_solvers$"*.
2) *Epochs*: Represents the number of times that the minimization in the algorithms is done. In the original implementation is 300.
3) *Theta*: Indirect bias threshold to select the neighbours for the repulsion set, neighbours with indirect bias grater than theta are considered for the repulsion set. This is set to 0.05 in the original implementation.
4) *Neighbours*: The number of neighbors considered for the repulsion set is determined by this parameter. In the original implementation, 100 neighbors are included in this set.
5) *Learning Rate*: This parameter dictates the learning rate employed by the optimizer throughout the optimization process. Originally this value is set to 0.01.
6) *Weights*: These represent the initial weights assigned to each function. In the original implementation, they are uniformly initialized as $[0.33, 0.33, 0.33]$

C. Comparing Algorithms

Our proposed methodology for a robust comparison of bias mitigation algorithms involves two main components. First, we perform hyper-parameter optimization by adapting the nested cross-validation technique from supervised learning. This choice is due to the advantages mentioned in Section III-A, where nested cross-validation enables robust model comparison, aligning with the goals of our methodology. Second, we leverage the results from this optimization to compare bias mitigation algorithms, not only by utilizing different sets of words but also by evaluating them at their optimal performance using the hyper-parameters that maximize bias reduction.

Given the similarities between bias mitigation algorithms and supervised learning algorithms, we propose adapting nested cross-validation to optimize the hyper-parameters of these algorithms.

To implement this methodology, we first select a set of word pairs to serve as our entire dataset. This dataset is analogous to the labeled data used in supervised learning. We then define a grid of hyper-parameters to be tested.

The nested cross-validation process involves dividing the word pair set into 10 folds, each containing bias definition (training) and target (testing) sets. Within each fold, the training portion is further subdivided into bias definition (train) and target (test) sets. We then iterate over the grid of hyper-parameters, testing each combination by performing the debiasing process and measuring the resulting bias. After testing all combinations, we select and store the combination that yields the best results. This optimal combination is then used to debias the entire training set of the fold and is evaluated on the test set.

By testing all combinations of hyper-parameters on the entire dataset, this approach provides more reliable results. The process is outlined in pseudo-code in Algorithm 1 and illustrated in Figure 2.

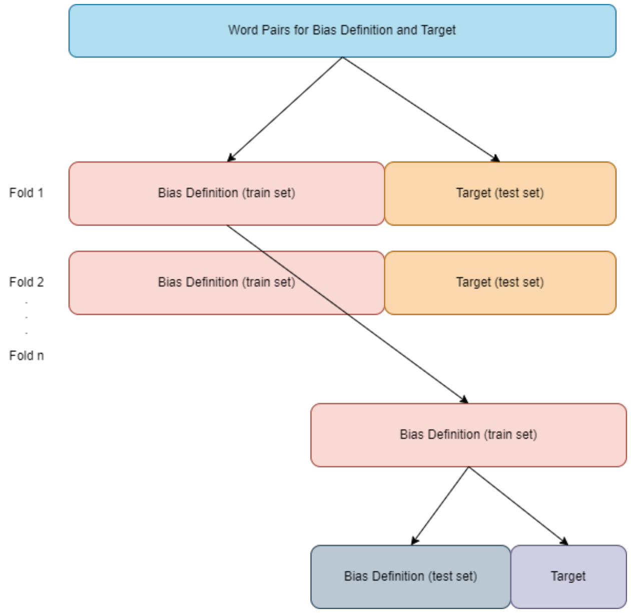
Word Pairs for Bias Definition and Target
Fold 1
Bias Definition (train set)
Target (test set)
Fold 2
Bias Definition (train set)
Target (test set)
.
.
.
Fold n
Bias Definition (train set)
Bias Definition (test set)
Target

Fig. 2: Diagram illustrating nested cross-validation for bias mitigation algorithms.

Algorithm 1 Nested cross-validation for Bias Mitigation

```
Define $grid$ of hyper-parameter combinations
Define set of $wordPairs$
Divide $wordPairs$ into $k$ $folds$ each with $train$ and $validation$ sets
Define $bestBias$ and $bestParam$
for fold in folds do
    Divide $train$ set of $fold$ into $train$ and $test$
    for $param$ in $grid$ do
        $algorithm.fit(train, param)$
        $algorithm.transform()$
        $bias \leftarrow algorithm.measureBias(test)$
        if $bestBias > bias$ then
            $bestBias \leftarrow bias$
            $bestParam \leftarrow param$
        end if
    end for
    $algorithm.fit(train, bestParam)$
    $algorithm.transform(validation)$
    $bias \leftarrow algorithm.measureBias()$
end for
```

After performing nested cross-validation, we use the bias measurements obtained from each outer fold to conduct the bias mitigation and measurement process. The process for each fold is executed using the optimal hyper-parameters identified during the cross-validation.

Our comparison method involves analyzing the results from each of the 10 folds, providing a comprehensive evaluation of each algorithm. By using subsets of the data for bias mitigation and measurement across these folds, we obtain more robust results that are not dependent on any single data set. This approach enables us to apply statistical tests to assess the significance of the results.

Hyper-parameter	**Values**
	HD
Solver	auto, full, arpack, randomized
	DHD
Solver	auto, full, arpack, randomized
	HSR
Alpha	0,10,30,60,80
	RAN
Neighbours	50,100,200
Theta	0.002,0.005,0.1

TABLE III: Hyper-parameters grids used for experiments.

We employ the corrected paired t-test [7] to compare the reduction of bias between a baseline model and a bias mitigation algorithm based on a chosen metric. This test is crucial because it addresses the limitations of the standard paired t-test when applied to cross-validated results. In cross-validation, the folds are not independent, which violates a key assumption of the standard paired t-test, leading to potentially misleading conclusions. The corrected paired t-test accounts for this lack of independence, providing a more accurate and reliable comparison of average performance statistics across cross-validated folds.

IV. Experiments

After defining our methodology, we conducted experiments to test its effectiveness. For this, we adopted the methodology proposed in [5], adhering closely to their experimental setup and utilizing the same word embedding model (GloVe-Wiki-Gigaword-300[1]). All of our experiments focused on gender bias, adopting the constraints they proposed and using the same word sets. We extended the word sets to define bias and target more comprehensively. Specifically, for our gender bias experiments, the word pairs used to define the target and bias sets included male and female group terms, such as: (she/he, woman/man, girl/boy, lady/gentleman, etc.). This extension increases the number of examples available for iteration[2].

To assess the optimal hyper-parameter combinations, we conduct experiments using each of the six bias measurement metrics described in Section II-A as optimization criteria[3]. Our goal is to identify the hyper-parameters that maximize the debiasing effect for each bias mitigation algorithm.

A. Grid Definition

The hyper-parameter grids used for each algorithm are presented in Table III. We briefly discuss each of these grids below.

[1] https://github.com/RaRe-Technologies/gensim

[2] The word sets and code used in our experiments for reproducibility purposes can be found at https://github.com/mzambrano1/Optimizing-Bias-Mitigation-A-Methodological-Approach-to-Hyperparameter-Tuning

[3] Due to space limitations, we only report results obtained with WEAT, RNSB, and RIPA as optimization criteria.

Algorithm	Δ Weat	Δ RND	Δ RIPA	Δ RNSB	Δ ECT	Δ WEATES
WEAT						
HD (Baseline)	-0.1 ± 0.031	-0.234 ± 0.102	-0.186 ± 0.035	-0.073 ± 0.024	0.14 ± 0.113	-0.325 ± 0.114
DHD	-0.033 ± 0.009 (↓)	-0.024 ± 0.011 (↓)	-0.035 ± 0.012 (↓)	-0.004 ± 0.012 (↓)	0.008 ± 0.026 (↓)	-0.056 ± 0.036 (↓)
RAN	-0.099 ± 0.037 (=)	-0.233 ± 0.101 (=)	-0.19 ± 0.036 (↑)	-0.074 ± 0.034 (=)	0.15 ± 0.123 (=)	-0.341 ± 0.129 (=)
HSR	-0.124 ± 0.032 (↑)	-0.022 ± 0.092 (↓)	-0.157 ± 0.036 (↓)	-0.042 ± 0.033 (↓)	-0.151 ± 0.082 (↑)	-0.229 ± 0.082 (=)
RNSB						
HD (Baseline)	-0.1 ± 0.031	-0.234 ± 0.102	-0.186 ± 0.035	-0.072 ± 0.026	0.14 ± 0.113	-0.325 ± 0.114
DHD	-0.033 ± 0.009 (↓)	-0.024 ± 0.011 (↓)	-0.035 ± 0.012 (↓)	0.002 ± 0.018 (↓)	0.008 ± 0.025 (↓)	-0.056 ± 0.036 (↓)
RAN	-0.096 ± 0.037 (=)	-0.234 ± 0.102 (=)	-0.19 ± 0.035 (↑)	-0.075 ± 0.033 (=)	0.154 ± 0.108 (=)	-0.351 ± 0.152 (=)
HSR	-0.124 ± 0.031 (↑)	-0.02 ± 0.09 (↓)	-0.156 ± 0.036 (↓)	-0.04 ± 0.031 (↓)	-0.148 ± 0.08 (↓)	-0.248 ± 0.095 (=)
RIPA						
HD (Baseline)	-0.1 ± 0.031	-0.234 ± 0.102	-0.186 ± 0.035	-0.072 ± 0.022	0.14 ± 0.113	-0.325 ± 0.114
DHD	-0.033 ± 0.009 (↓)	-0.024 ± 0.011 (↓)	-0.035 ± 0.012(↓)	-0.003 ± 0.013 (↓)	0.008 ± 0.026 (↓)	-0.056 ± 0.036 (↓)
RAN	-0.096 ± 0.041 (=)	-0.233 ± 0.101 (=)	-0.191 ± 0.034 (↑)	-0.073 ± 0.034 (=)	0.144 ± 0.133 (=)	-0.358 ± 0.158 (=)
HSR	-0.125 ± 0.03 (↑)	-0.02 ± 0.09 (↓)	-0.157 ± 0.036 (↓)	-0.039 ± 0.035 (↓)	-0.143 ± 0.077 (↓)	-0.261 ± 0.072 (=)

TABLE IV: Mean change in bias produced by Bias Mitigation Algorithms across 10 runs of nested cross-validation for three optimization metrics: WEAT, RNSB, and RIPA. A more negative result indicates a greater reduction in bias in the final model, except in the case of the ECT metric, where a more positive result signifies less bias. Next to each result, an arrow or equal sign indicates whether the outcome is significantly better, worse, or equal to the baseline (Hard Debias).

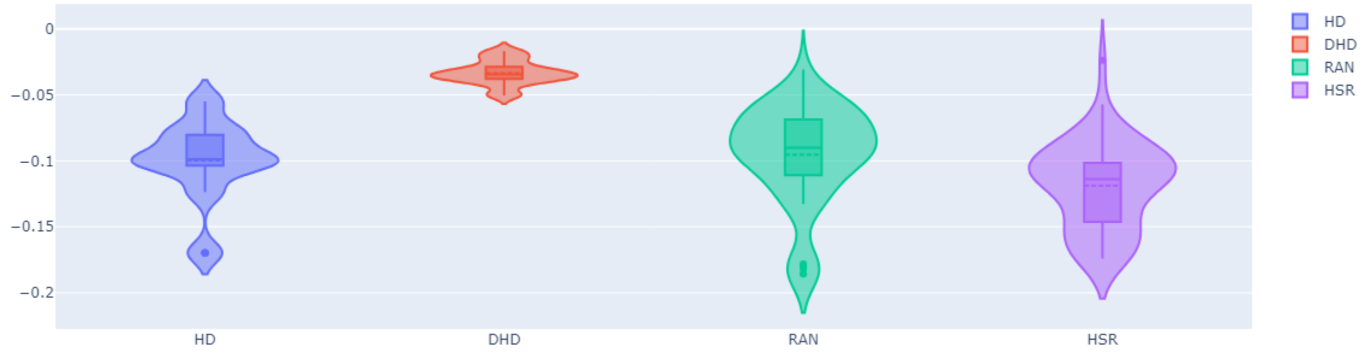

Fig. 3: Violin Plot of WEAT results for all experiments.

1) Hard Debias and Double Hard Debias: As the only hyper-parameter to adjust in both algorithms is the solver for the PCA, we test each of the possible values available for this hyper-parameter.

2) Half Sibling Regression: For Half Sibling Regression's single hyper-parameter, we utilize the values presented in Table III. This allows us to explore the parameter's impact on the process, starting with a simple linear regression and incrementally increasing the value.

3) Repulsion Attraction Neutralization: For Repulsion Attraction Neutralization, we include the hyper-parameters "neighbours" and "theta" in the grid. These parameters are used to construct the repulsion set and interact with each other, defining the size of this set. The values are shown on Table III.

B. Results

After conducting hyper-parameter optimization on all four bias reduction algorithms mentioned in Section II-B, using all six bias measurement metrics as optimization criteria, we gain insights in three key areas: (1) a comparative analysis of the algorithms, (2) an understanding of how the choice of optimization metric influences the results, and (3) identification of the hyper-parameters that optimize the performance of each algorithm.

Table IV presents the mean results obtained for each algorithm after applying the proposed methodology, along with the corresponding standard deviation. Bias is evaluated in each run using the optimal hyper-parameters identified, with measurements based on the metrics described in Section II-A. Each metric serves as an optimization criterion, and the table displays results for three of these metrics: WEAT, RNSB, and RIPA. Figure 3 shows a violin plot of the WEAT metric results across all runs, using all metrics as optimization criteria, for all four algorithms.

The results in Table IV involve calculating the difference in bias within the model before and after applying the mitigation. In this context, a more negative value indicates greater bias mitigation, except for the ECT metric, where the interpretation is reversed. An equal or arrow symbol is placed next to each result to indicate whether it is significantly better, worse, or equal to the baseline, according to a corrected paired t-test [7] with a significance level of 0.05. This correction is necessary because, as noted by Nadeau and Bengio (2003) [7], the common t-test is not suitable for these experiments due to the assumption of independent samples, which does not hold in this case. Hard Debias is used as the baseline because it was the first algorithm proposed.

From these results, it is evident that none of the more recently proposed algorithms outperform Hard Debias. Both Double Hard Debias and Half-Sibling Regression generally perform worse, while Repulsion Attraction Neutralization is statistically equivalent to Hard Debias. This highlights that despite efforts to improve bias mitigation over the years, the initial Hard Debias algorithm remains as effective as any subsequent approaches.

In addition, Figure 3 shows that although Double Hard Debias (DHD) is the least effective in mitigating bias, it produces the most stable results according to the WEAT metric. This is consistent with the number of hyper-parameters in the method, but it is also noteworthy because HD has the same hyper-parameter but a significantly larger variance.

Another insight gained from these results is that the choice of optimization metric has minimal impact, as the metrics appear to guide the optimization process similarly. This implies that all metrics offer comparable information about the bias of the embedding models and can be used interchangeably.

In Table V we present the results showing the most frequently chosen optimal hyper-parameter combination for each algorithm. For HD and DHD, the best hyper-parameter is "auto" for the solver, which is also the default value. In contrast, for HSR and RAN, the best hyper-parameters differ from those proposed by the authors. Interestingly, for HSR, the optimal value for α is 0, which effectively turns the regression used by the algorithm into a standard linear regression, contrary to what was proposed by the authors of the method [13], who suggested a value of 60.

Algorithm	Hyper-parameter	Value
HD	solver	auto
DHD	solver	auto
HSR	alpha	0
RAN	theta,neighbour	0.02,200

TABLE V: Best hyper-parameters combination obtained for each algorithm.

V. Conclusions

In this paper, we presented a methodology for robustly comparing word embedding bias mitigation algorithms by exploiting the similarities between supervised learning and bias mitigation algorithms, incorporating techniques such as nested cross-validation from supervised learning.

This methodology enables a robust comparison of bias mitigation algorithms by addressing the variability introduced by different word sets and hyper-parameters. It also facilitates statistical analysis to assess the significance of the results.

Our experiments revealed that despite advancements in bias mitigation algorithms, none significantly outperforms the original Hard Debiasing approach proposed by Bolukbasi et al. [1].

These results align with those reported in [5], which also showed that when algorithms are compared in controlled settings, the differences in performance are not substantial.

References

[1] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[2] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[3] C. Sweeney and M. Najafian, "A transparent framework for evaluating unintended demographic bias in word embeddings," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 1662–1667. [Online]. Available: https://aclanthology.org/P19-1162

[4] T. Wang, X. V. Lin, N. F. Rajani, B. McCann, V. Ordonez, and C. Xiong, "Double-hard debias: Tailoring word embeddings for gender bias mitigation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5443–5453. [Online]. Available: https://aclanthology.org/2020.acl-main.484

[5] F. Bravo-Marquez and M. J. Zambrano, "Unpacking bias: An empirical study of bias measurement metrics, mitigation algorithms, and their interactions," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia, 2024, pp. 17 154–17 164. [Online]. Available: https://aclanthology.org/2024.lrec-main.1490

[6] M. Stone, "An asymptotic equivalence of choice of model by cross-validation and akaike's criterion," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 44–47, 1977.

[7] C. Nadeau and Y. Bengio, "Inference for the generalization error," in *Advances in Neural Information Processing Systems*, vol. 12, 1999.

[8] N. Garg, L. Schiebinger, D. Jurafsky, and J. Zou, "Word embeddings quantify 100 years of gender and ethnic stereotypes," *Proceedings of the National Academy of Sciences*, vol. 115, no. 16, pp. E3635–E3644, 2018.

[9] K. Ethayarajh, D. Duvenaud, and G. Hirst, "Towards understanding linear word analogies," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 3253–3262. [Online]. Available: https://aclanthology.org/P19-1315

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

[11] P. Badilla, F. Bravo-Marquez, and J. Pérez, "Wefe: The word embeddings fairness evaluation framework," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 430–436. [Online]. Available: https://doi.org/10.24963/ijcai.2020/60

[12] V. Kumar, T. S. Bhotia, and T. Chakraborty, "Nurse is closer to woman than surgeon? mitigating gender-biased proximities in word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 486–503, 2020.

[13] Z. Yang and J. Feng, "A causal inference method for reducing gender bias in word embedding relations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9434–9441.

[14] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*. Pearson, 2018.

[15] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.

[16] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.