UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IMPLEMENTACIÓN DE ALGORITMOS DE REDUCCIÓN DE SESGO EN WEFE

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN COMPUTACIÓN

MARÍA JOSÉ ZAMBRANO BURROWS

PROFESOR GUÍA:
FELIPE BRAVO MÁRQUEZ

PROFESOR CO-GUÍA:
PABLO BADILLA TORREALBA

MIEMBROS DE LA COMISIÓN:
AIDAN HOGAN
MAURICIO CERDA VILLABLANCA

Este trabajo ha sido parcialmente financiado por FONDECYT 11200290

SANTIAGO DE CHILE
2022

Resumen

Se ha demostrado en los últimos años que los modelos de *word embeddings* logran captar los sesgos presentes en los corpus de los que son entrenados, aprendiendo relaciones con estereotipos de género, raciales, entre otros. A modo de abordar el problema del sesgo en modelos de *word embeddings* se han propuesto distintas métricas para cuantificarlo y algoritmos para mitigarlo.

En el departamento de Ciencias de la Computación de la Universidad de Chile se desarrolló *The Word Embedding Fairness Evaluation Framework (WEFE)*, una librería de código abierto para *Python*. WEFE tiene como objetivo unificar las métricas de cuantificación de sesgo y los algoritmos de mitigación de este.

En esta memoria se tiene como objetivo la implementación de tres nuevos algoritmos de mitigación de sesgo en WEFE, como también la experimentación con ellos. Para lograr este propósito es necesario homogeneizar los algoritmos para que puedan ser usados de una manera estándar.

La estandarización de los algoritmos en WEFE se logra adaptándolos a la interfaz *fit-transform* de *Scikit-learn*. Una vez integrados los métodos de mitigación de sesgo en la librería se realizan una serie de experimentos de manera de corroborar su correcta implementación y verificar empíricamente su efecto en los modelos de embeddings, tanto en el sesgo como en su funcionalidad.

Los resultados de experimentales muestran que los algoritmos fueron implementados de manera correcta en WEFE, como también dan cuenta de las diferentes reducciones de sesgo derivadas de la aplicación de estos. Finalmente se comprobó que las operaciones de mitigación de sesgo no tienen un efecto negativo en el funcionamiento de los *word embeddings*, es más, hay casos en los que lo mejora.

Con la implementación de los algoritmos, como también los experimentos realizados con ellos, se lograron los objetivos propuestos para esta memoria. Esto da como resultado una nueva versión de WEFE, que ahora incluye cinco algoritmos de reducción de sesgo.

Abstract

In recent years, it has been shown that word embeddings models are able to capture the bias present in the corpora from which they are trained, learning stereotyped relations. To address the problem of bias in word embeddings models, different metrics to quantify it and algorithms to mitigate it have been proposed.

The Word Embedding Fairness Evaluation Framework (WEFE), an open source library for Python, was developed at the Computer Science Department of the University of Chile. WEFE aims to unify the bias measurement metrics and bias mitigation algorithms.

The objective of this work is to implement three new bias mitigation algorithms in WEFE, unifying them so that they can be used in a standard way, as well as to experiment with them. The standardization of the algorithms is achieved by adapting them to Scikit-learn's fit-transform interface.

After integrating the bias mitigation methods into WEFE, we perform a series of experiments in order to check their correct implementation, compare them with one another and verify that mitigating the bias does not affect the functionality of the word embeddings models.

The results we obtain from the experiments show that the algorithms were implemented correctly in WEFE. They also account for differences in the decrease of bias between them. Finally, we found that the bias mitigation operations do not have a negative effect on the performance of the embeddings, in fact, there are cases where it improves it.

With the implementation of the algorithms, as well as the experiments carried out with them, we achieved the the objectives proposed for this work. These results in a new version of WEFE, which now includes five bias reduction algorithms.

It's gonna be legen... wait for it. . .

Agradecimientos

En primer lugar, agradecer a mis padres, Marco y Cecilia por todo el cariño y apoyo que me han dado desde siempre. Gracias por hacer todo esto posible y por soportarme más que ninguna otra persona. Gracias también a mis hermanos, el Mati y la Bea, los cabros chicos que siempre han estado por ahí molestando de alguna forma.

Agradezco enormemente a mi pololo, Joaquín, que ha tenido toda la paciencia del mundo al soportarme estos años, me ha apoyado y acompañado en todo lo que se me ha ocurrido. Muchas gracias por todo.

Mil gracias a mi profes, Felipe y Pablo, que me dieron esta gran oportunidad y la confianza que tuvieron mi e hicieron esto posible.

A los Rodríguez Christian, que me adoptaron y acogieron en mis primeros años en la capital, les agradezco por todo el cariño. Gracias por darme un hogar y soportarme por tanto tiempo.

Por último, pero no menos importante, agradezco a mis amigos, los cabros de la dieta milagrosa que son muchos para nombrarlos a todos, pero mención honrosa a la Consu que se dio la lata de leer todo esto y corregirme las faltas de ortografía. Gracias por acoger a esta sureña perdida en la capital, por todos los almuerzos y los juegos de cartas, lejos lo mejor de estos años.

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Natural Language Processing (NLP) is an area of study of computer sciences that focuses on the design and analysis of computational algorithms and models that allows producing and process natural human language [11]. To achieve these objectives it is necessary to represent the human vocabulary in a way that can be understood by computers. Word embeddings are some of the most used models in the field to solve this task.

Word embeddings are models focused on representing words in vector form. There are several ways to obtain these type of representations, one of them being *Skipgram* [23]. This technique consists in training a neural network to predict the words that surround a central word that shifts along the training corpus. This training process is done over large text corpora.

The representations learned by word embeddings allow word vectors from words used in similar contexts to have less distance between them than vectors of words appearing in very different contexts. This ability that word embeddings have of learning from context is such that they are able to compute word pair relationships from vector operations such as country-capital or that “king” is to “man” as “queen” is to “woman”, which is quite desirable in text representation models.

It has been found that some word embedding models learn relationships such as “man” is to “computer programmer” as “woman” is to “homemaker” [7], or also that “nurse” is closer to “woman” than “surgeon” [21]. These relationships evidence the existence of bias within the word embedding models.

Examples such as Microsoft’s Twitter bot that had to be shut down after only 24 hours after learning racist and gender-discriminatory language [3], or Amazon’s recruiting tool that only selected men [2] show that bias is an important issue in machine learning that should be taken to consideration.

It is due to situations such as those previously mentioned that bias in NLP and particular in word embeddings is a topic of great research interest. In the past few years, several teams have studied bias through different metrics such as Word Embedding Association Test [18], Relative Norm Distance [14], Relative Negative Sentiment Bias [30]. As well as different mitigation algorithms, also known as debias algorithms, such as Hard Debias [7] Double Hard Debias [31], Repulsion Attraction Neutralization [21] among others, resulting in fairer representations of the vocabulary.

1.2 Description of the Problem

Word Embedding Fairness Evaluation (WEFE) framework [5] was developed at the Department of Computer Science of the University of Chile, which with more than 400 downloads in the last month is being widely used [1]. The original objective of WEFE was to generalize different bias measurement metrics, making it possible to use them in a standard way and to compare them with each other. Later, WEFE was expanded to include bias mitigation methods.

One of the limitations of WEFE lies in the fact that so far it only allows basic bias mitigation operations in word embedding models, including two bias mitigation algorithms. However, one of the library's objectives is to implement state of art bias mitigation algorithms. The main problem with the integration of these methods is that each algorithm has its own functioning and inputs, this is why it is not direct to include them in WEFE in a standardize way.

1.3 Objectives

1.3.1 General Objectives

The main objective of this work is to extend the WEFE library so it includes three new bias mitigation algorithms: Double Hard Debias, Half Sibling Regression and Repulsion Attraction Neutralization. This way the library will be able to expand its reach in the bias mitigation subject, incorporating new bias mitigation algorithms and compare them with the existing metrics. The result of meeting this objective is to make available to the community a more complete library, as well as conducting a case study that compares the performance of the different bias mitigation algorithms using the original case study from WEFE [5].

1.3.2 Specific Objectives

1. Design a method to unify the algorithms to be implemented so they can be used in a standard way, like the metrics already included in WEFE.
2. Study different word embeddings bias mitigation approaches to fully comprehend their functioning so the integration into the framework is done properly.

3. Continue the work done in the Word Embeddings Benchmark to obtain a functional version of the software and to have a valid tool to evaluate the performance of word embeddings in classical tasks.
4. Compare the performance of different word embedding models (using the updated version of WEB), after applying bias mitigation algorithms to them, to make sure their functioning is not affected by the debias.
5. Compare the effectiveness of the different bias mitigation algorithms implemented during this work using the metrics included in WEFE.

1.4 Bias Mitigation in WEFE

The inclusion of the bias mitigation algorithms into WEFE is done by adapting them to fit in the "fit-transform" interface inherited from Scikit-learn and to receive target and ignore word sets as parameters. This way the debias process is done in two main steps: "fit" where all the necessary transformations that will mitigate bias are learned and "transform" where the transformations are applied over the model.

The algorithms are also adapted to receive as parameters "target" and "ignore" word sets. These sets indicate the words to which the debias is applied and the words that will not be included in the process respectively.

1.5 Results

As a result of the work done, we were able to expand WEFE so that it now implements a more complete bias mitigation module including 3 new bias mitigation algorithms: "'Double Hard Debias", "Half Sibling Regression" and "Repulsion Attraction Neutralization". WEFE is now ready for the release of a new version that incorporates the recently implemented algorithms.

In addition to the above, we were able to perform several experiments, validating the correct implementation of the algorithms, verifying that the algorithms do reduce bias in the word embedding models, including a comparison of their performance and checking that the debias process does not affect the functioning of the models. This way we were able to achieve all of the objectives proposed in Section 1.3.

1.6 Outline

The rest of this work is organized as follows:

- Chapter 2: Background and Related Work. We briefly introduce natural language processing, word embedding models, and bias in these models. We also review the related work done on the field describing bias measurement metrics, bias mitigation algorithms and other relevant frameworks for the work.

- Chapter 3: Implemented Methods. In this chapter we present the similarities found among the algorithms, how they are standardized and included into WEFE and a description of the update done to WEB.
- Chapter 4: Experiments. In this chapter we described the experiments performed with the implemented methods and present the results obtained.
- Chapter 5: Conclusions and Future Work. Finally, we review all the work, discuss the achievement of the established objectives and propose future work.

Chapter 2

Background and Related Work

In this chapter we intend, first, to introduce the reader to the knowledge areas addressed in this work, such as natural language processing, word embeddings models and bias. Secondly, we aim to go through the related work existing in the area of bias in word embeddings models.

This chapter is structured as follows: First, in Section 2.1 the natural language processing field is briefly introduced. Section 2.2 describes word embeddings, Section 2.3 focuses on describing bias in word embedding models, how to measure it and bias mitigation methods. Finally, Section 2.4 introduces the Word Embedding Fairness Evaluation Framework.

2.1 Natural Language Processing

Natural Language Processing, according to Goldberg [15], is the field of designing methods and algorithms that take as input or produce as output unstructured, natural language data. In practical terms, NLP focuses its efforts on solving well-defined and limited tasks such as translation, topic-classification, part-of-speech tagging, among several others [4].

Solving these tasks represents a great challenge, since natural language is not an easy thing to work with; even for humans it can sometimes be difficult to understand. This is mainly because of some challenging properties that language possesses:

- **Discreteness:** It is not possible infer the relationship between two words from the letters that compose them. Meaning that two words could have a strong relationship, for example “fork” and “spoon”, but the letters that compose them are completely different.
- **Compositionality:** In order to understand a sentence it is necessary to analyze it as a whole and not the words that compose it individually, because the meaning of the sentence goes beyond the words that compose it.
- **Sparseness:** There is practically an infinite number of combinations that can be done with letters, thus there is, as well, practically an infinite number of meanings.

An essential thing for NLP is how to represent the language so it can be understood by computers and used to resolve the mentioned tasks. Over the years different approaches have been taken to do so. Some examples are One Hot representations, Word Context Matrices, Word Embeddings, etc. We focus in Word Embeddings models as representations of language.

2.2 Word Embeddings

Word embeddings or distributed representations are models that encode the meaning of words in dense vectors, such that words with similar definitions tend to have similar vectors as representation. These vectors are based in the distributional hypothesis, stating that words that appear in similar context tend to have similar meanings [16].

Word embeddings are trained using neural networks over large corpora of documents. During this process the semantics of the words is distributed along the vectors (giving them the name of distributed representations). Because of the procedure the dimensions of the vectors are not interpretable [4].

In the following subsections we will discuss tasks word embeddings can solve, the training process for word embeddings, pre-trained models and how their performance is evaluated.

2.2.1 Tasks

We describe several tasks that word embeddings are able to solve:

- **Similarity**: It is possible to measure the similarity between two words measuring the similarity between their vectors: the more similar the vectors are, the more similar the words. This can be done with any vector similarity metric, such as cosine similarity.
- **Analogy**: Word analogies can often be solved with vector arithmetic; this way it is possible to find that "queen" is the answer to "king is to ? as man is to woman".[12]
- **Categorization**: Using clustering algorithms, such as K-means, it is possible to find clusters that are expected to group words according to some category.

2.2.2 Training process

Several methods have been developed for training word embeddings models, such as Skipgram [23] ,Words2vec [24], Fasttext [6], Glove [26], among others. We next explain the Skipgram model in order to exemplify the training process of word embeddings models.

The Skipgram models consists in training a neural network with one hidden layer and no activation function to predict the words surrounding a central words.

To achieve this, a vocabulary of $10,000$ words is extracted from a corpus, which are represented as one-hot encoded vectors. This consists in generating, for each word, a vector the size of the vocabulary where each dimension is related to a specific word of the vocabulary, resulting in a vector that contains only zeros in its dimensions, except in the one that represents the corresponding word.

This way the neural network to be trained receives as input central words, in one-hot encoding, with the objective of predicting the surrounding words. In the hidden layer each neuron has a total of 300 weights without activation function and in the output layer each neuron uses a softmax function obtaining a vector of $10,000$ dimensions with the probabilities that each word has to surround the central word.

The corpus is traversed in its entirety during training, predicting from the central word and adjusting the weights of the neurons in the hidden layer by using backpropagation. When the process is finished the matrix contained in the hidden layer is used as a word embedding.

2.2.3 Pre-trained models

Once a word embedding models is trained, it can be used to complete different tasks, as those previously explained.

When a word embedding model has already been trained it can be downloaded and used for multiple purposes. There are libraries that implement different algorithms that enable their training. A library that allows both of these things, training word embeddings models and download trained models, is Gensim[1], an open source python library for topic modelling, document indexing and similarity retrieval with large corpora.

2.2.4 Evaluation

The principal objective of evaluating word embedding models is to verify that they are able to capture syntactic and semantic properties of words satisfactorily. To do so there are different ways of evaluating pre-trained word embedding models, which are divided into two approaches: intrinsic evaluation and extrinsic evaluation.

Intrinsic evaluation measures the quality of the embedding based on how they perform on different tasks such as categorization, similarity, analogy. This is done over datasets created by human experts.

On the other hand, extrinsic evaluation is performed using the models in different NLP tasks, such as sentiment analysis, part of speech tagging, etc. The idea behind this is to see how well the tasks are solved with the use of the word embeddings models.

For evaluating pre-trained word embeddings models a Python library was developed, Word Embeddings Benchmark, that we explain next.

[1] `https://radimrehurek.com/gensim/`

Word Embeddings Benchmark

Word Embeddings Benchmark (WEB) is an open source library implemented in Python, that focuses on providing simple methods to evaluate word embedding models.

WEB uses different labeled datasets built for common NLP tasks to evaluate word embedding models. These sets are divided into three categories: categorization, similarity, analogy. For a given task the model to evaluate is used to complete it and then the results are compared with the correct ones in the dataset. A score is then is assigned according to how well the model did resolving the task. The higher the score, the better the model solved the task.

This library can be found in its own Github[2] repository, but it has not been updated in over two years, thus it has some deprecation issues. [19]

2.3 Bias in Word Embeddings

The Oxford dictionary defines bias as *"a strong feeling in favour of or against one group of people, or one side in an argument, often not based on fair judgement"* [10]. It is no surprise that we coexist with bias in our daily life, whether it is gender, racial, political, etc. and this is reflected in what we write and what we read, on social media, news articles, etc.

Recent works demonstrate that word embeddings, among other methods in machine learning, capture common stereotypes because these stereotypes are likely to be present, even if subtly, in the large corpora of training texts [14]. This leads to an unfair representation of the language causing undesirable relations, for example, that men are doctors and women are nurses [21].

Therefore, in recent years, bias in word embedding models have been of great interest in NLP research. Here we present some metrics developed to measure bias and algorithms proposed to reduce it.

2.3.1 Bias Measurement

In recent years, several studies have proposed different metrics with the aim of quantifying the bias present in word embeddings models. Some examples of these metrics are Word Embedding Association Test (WEAT) [18], Relative Norm Distance (RND) [14], among others. Even though all the metrics focus on quantifying bias, each metric proposes its own way of measuring it. This means that the results obtained by the different metrics are not directly comparable, which makes it very complex to standardise the measurement of bias in pre-trained word embeddings models.

The following is a brief explanation of the bias quantification metrics relevant for this work.

[2] https://github.com/kudkudak/word-embeddings-benchmarks

Word Embedding Association Test (WEAT)

Proposed by Caliskan et al. [18] this metric focuses on quantifying the level of association between two pairs of word sets, receiving as input two objective sets, such as: $\{programmer, engineer, scientist...\}$ and $\{nurse, teacher, librarian...\}$ and two attributes sets such as: $\{man, male...\}$ and $\{woman, female...\}$.

For X and Y target word sets, A and B attributes word sets, the WEAT metric is defined as follows:

$$s(X, Y, A, B) = \sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B)$$

$$s(w, A, B) = mean_{a \in A} cos(\vec{w}, \vec{a}) - mean_{b \in B} cos(\vec{w}, \vec{b})$$

with $cos(\vec{a}, \vec{b})$ the cosine of the angle between vectors $\vec{a}$ y $\vec{b}$.

This way the more positive the result is the more related is target set X to attribute set A and Y to B. On the other hand, the more negative the result the more related is set X to B and Y to A, being the ideal result 0, showing no sign of bias between the sets.

WEAT Effect Size (WEAT-ES)

Proposed by Caliskan et al. [18] this metric represents a normalized measure of how separated the two distributions of associations between the target and attribute are, being the ideal value (meaning no bias) 0 and the higher the value the more biased the model is. WEAT-ES is defined as follows:

$$\frac{mean_{x \in X} s(x, A, B) - mean_{y \in Y} s(y, A, B)}{std - dev_{w \in X \cup Y} s(w, A, B)}$$

Where X,Y are target sets and A,B are attribute sets, like the ones defined for WEAT.

Relative Norm Distance (RND)

Proposed by Garg et al. [14], this metric captures the relative strength of association between a set of neutral words and two sets of words. RND is define as the following:

$$RND = \sum_{v_m \in M} ||v_m - v_1||_2 - ||v_m - v_2||_2$$

where M is the set of neutral words and v_i is the average vector for word sets 1 and 2.

The more positive the result is, the more related are the neutral words to word set number one, and on the contrary, the more negative the result is the more related they are to word set number 2, being the ideal result 0.

Relative Negative Sentiment Bias (RNSB)

Proposed by Sweeney [30], RNSB trains a binary classifier $f^*(x_i)$ over gold standard labeled sentiment words, (x_i, y_i), where x_i is a word vector from a possibly biased word embedding model and y_i is the negative or positive label assigned to the word. Then, for a set $K = \{k_1, ..., k_t\}$ of word vectors from a particular demographic group (i.e. national origin, religion, gender etc) a set P is defined, that can be interpreted as a probability distribution for k_i, as follows:

$$P = \{\frac{f^*(k_i)}{\sum_{i=1}^{t} f^*(k_i)}, ..., \frac{f^*(k_t)}{\sum_{i=1}^{t} f^*(k_i)}\}$$

The idea behind RNSB is that the more similar P is to a uniform distribution, the less biased the word embedding model is. The metric is defined as:

$$RNSB(P) = D_{KL}(P||U)$$

where U is the uniform distribution and D_{KL} is the Kullback-Leibler divergence, which measures the similarity between two probability distributions.

This means that RNSB represents the similarity between the set P and the uniform distribution; thus the lower the result the more fair the word embedding model is.

Relational Inner Product Association (RIPA)

Ethayarajh [13] proposes RIPA for a word vector $\vec{v} \in V$ with respect to a relation vector $\vec{b} \in V$, which is defined as:

$$\beta(\vec{w}, \vec{b}) = \langle \vec{w}, \vec{b} \rangle$$

Where $\vec{b}$ is defined as the first principal component of $\{\vec{x} - \vec{y} \mid (x, y) \in S\}$, with S a non-empty set of ordered word pairs, and can be written as $\vec{b} = \frac{\vec{x}-\vec{y}}{||\vec{x}-\vec{y}||}$ when a single word pair defines the association.

For RIPA a value of 0 indicates no bias and the higher the value obtained, the more biased the word embedding model is.

Embedding Coherence Test (ECT)

ECT proposed by Dev and Phillips [**?**] focuses on measuring how related a set of professions are to gendered word pairs. To achieve this they define a set $P = \{p_1, p_2, ..., p_k\}$ of professions and a set ε of gendered pairs of words. Then, for all the word pair $(e_j^+, e_j^-) = E_j \in \varepsilon$ two means are computed:

$$m = \frac{1}{|\varepsilon|} \sum_{E_j \ \varepsilon} e_j^+ \quad s = \frac{1}{|\varepsilon|} \sum_{E_j \ \varepsilon} e_j^-$$

Next, the cosine similarity is calculated for both m and s to all the words $p_i \in P$, obtaining two vectors $u_m, u_s \in \mathbb{R}^k$. Then, the coordinates of these vectors are replaced by its rank in order to finally compute the Spearman correlation to words in P. The closer the result is to 1, the less bias.

Types of Bias

Bolukbasi et al. [7] define two types of bias present in word embedding models: Direct and indirect bias, as a way of measuring bias. We next briefly explain both of them.

Direct bias measures how far from the bias direction are a set of N neutral words. Higher values of direct bias indicate more biased models. Direct bias is defined as follows:

$$DirectBias_c = \frac{1}{|N|} \sum_{w \in N} |cos(\vec{w}, g)|^c$$

where c is a parameter to determine how strict is the bias measurement and g is the bias direction, a vector that captures bias in the embedding,.

Indirect bias measures how associated two words (w and v) are to bias, high values of indirect bias indicate strong association due to bias. Indirect bias is defined as follows:

$$\beta(w, v) = (w \cdot v - \frac{w_\perp \cdot v_\perp}{||w_\perp||_2 ||v_\perp||_2})/w \cdot v$$

where g is the bias direction, $w_\perp = w - w_g$ and $w_g = (w \cdot g)g$.

2.3.2 Bias Mitigation

To obtain less biased word embeddings, different algorithms have been proposed, such as Hard Debias [7], Double Hard Debias [31], Repulsion, Attraction Neutralization [21], among others. These algorithms aim to mitigate bias in pre-trained word embeddings by focusing on adjusting the vectors in order to reduce the distances between words that are normally associated with different types of biases (e.g., gender, race, religion).

Each of the mentioned algorithms works in a different way, receiving different inputs and performing its own operations to achieve its goal. Each algorithm is implemented in its own repository, independently and their links can be found in Annexed B.1.

Next, we explain the bias mitigation algorithms relevant to this work.

Hard Debias (HD)

This method proposed by Bolukbasi et al. [7] reduces bias in word embeddings models through geometric operations following the next steps:

- Identify the bias subspace using a set of word pairs that define the bias, such as "*man*" and "*woman*".
- Neutralize the bias subspace by calculating the projection of the embedding on the bias subspace and then subtracting it from the embedding. This is applied to a set of neutral words that should not contain bias.
- Equalize the embeddings with respect to the bias direction. This step is applied to word pairs, such as the ones to identify the bias subspace, and it distributes both words at the same distance of the bias direction.

Multiclass Hard Debias (MHD)

Manzini et al. [22] proposed Multiclass Hard Debias, a generalization of Hard Debias that allows performing debias to word embedding models in multiple bias criteria. To do this they identify the bias subspace in a multiclass setting and then simply perform Hard Debias.

Repulsion, Attraction, and Neutralization (RAN)

RAN is a bias mitigation method proposed by Kumar et al [21]. With this method they claim not to only eliminate the bias present in word vectors, but also to alter the spatial distribution of its neighbours vectors achieving a bias-free setting while maintaining minimal semantic offset. All of this is achieved by creating a transformation $\{\vec{w_i}\}_{i=1}^{|V|} \rightarrow \{\vec{w_i'}\}_{i=1}^{|V|}$ that minimises the stereotypical bias with minimal semantic offset. This transformation bases its operations on:

1. Repelling embeddings from neighbours with a high value of indirect bias (indicating a strong association due to bias), to minimise the bias based illicit associations.
2. Attracting debiased embeddings to the original representation, to minimise the loss of semantic meaning.
3. Neutralizing the bias direction of each word, minimising its bias to any particular group.

For the debiasing process, two sets of words are defined

- **Preserved set** (V_p): Words where gender carries semantic importance (e.g., beard, bikini), therefore the debias process is not applied to them to preserve the semantic importance.
- **Debias set** (V_d): Words which are not present in (V_p). Expected to be gender neutral, hence they are subjected to the debiasing process. This sets includes gender stereotypical words (e.g., nurse, warrior) and gender neutral words (e.g., table, keyboard).

The method consists in a multi-objective optimization. For each word $w \in V_d$ and its vector $\vec{w}$ its debias counterpart $\vec{w_d}$ is found by solving the optimization problem:

$$argmin(F_r(\vec{w_d}), F_a(\vec{w_d}), F_n(\vec{w_d})) \tag{2.1}$$

The optimization is performed by formulating a single objective:

$$F(\vec{w_d}) = \lambda_1 F_r(\vec{w_d}) + \lambda_2 F_a(\vec{w_d}) + \lambda_3 F_n(\vec{w_d}) \tag{2.2}$$

where λ_i, hyperparameters of the method, are weights that determine the relative importance of one function (repulsion, attraction, neutralization) over another with the restriction $\sum_i \lambda_i = 1$ .F_i are the repulsion, attraction and neutralization functions. Each of these functions has its own purpose and they are defined as follows:

- **Repulsion**: Defines a repulsion set S_r and aims to reduce the unwanted semantic similarity between $\vec{w_d}$ and S_r
$$F_r(\vec{w_d}) = \frac{\sum_{n_i \in S_r} |cos(\vec{w_d}, \vec{n_i})|}{|S_r|}$$
- **Attraction**: Attracts $\vec{w_d}$ to $\vec{w}$, aiming to minimise the loss of semantic and analogical properties.
$$F_a(\vec{w_d}) = \frac{|cos(\vec{w_d}, \vec{w}) - 1|}{2}$$
- **Neutralization**: Aims to minimise the bias towards any particular gender. Represents the absolute value of dot product of word vector $\vec{w_d}$ with the gender direction $\vec{g}$ (as described in Hard Debias [7]).
$$F_n(\vec{w_d}) = |cos(\vec{w_d}, \vec{g})|$$

Double Hard Debias (DHD)

Wang et al [31] posit that word frequency in the training corpora can twist the gender direction and limit effectiveness of Hard Debias [7], which makes the assumption that it is possible to identify and isolate gender direction. To correct this they propose DHD. In addition to applying HD to the word embeddings, they subtract the frequency direction that distracts the gender subspace. This is done as follows:

- First, word embeddings are projected into an intermediate subspace by subtracting component(s) related to word frequency. Mu and Viswanath [25] say that the dominant directions of word embeddings encode word frequency, so the components are obtained by performing a Principal Components Analysis over the whole set of vectors.
- Then Hard Debias is applied to these purified embeddings to mitigate gender bias.

The debias process is applied to the k most biased words, with k a hyperparameter of the method. That is the $\frac{k}{2}$ words that are closer to "he" and the $\frac{k}{2}$ closer to "she". A defined a set of words is excluded from the debias process, specifically words in which gender carries semantic information.

Half Sibling Regression (HSR)

Yang [32] proposed this method based on a confounding-noise-elimination method [28]. HSR proposes to learn spurious gender information via causal inference by utilizing the statistical dependency between gender-biased word vectors and gender definition word vectors. The learned spurious gender information is then subtracted from the gender-biased words per the following equation:

$$V'_N := V_N - G.$$

G is obtained by predicting non-gender definition word vectors using the gender-definition word vectors.

$$G := E[V_N|V_D]$$

And the word sets are defined as:

- **Gender Definition (V_D):** Words associated with gender by definition, such as mother and father.
- **Non gender definition(V_N):** Words not associated with gender such as doctor and nurse.

The debias is performed in three majors steps. First, a Ridge Regression [17] is performed to compute a weight matrix W. The matrix is then used to predict the gender information using the gender definition word vectors. Finally, the gender information is subtracted from the non-gender definition word vectors.

$$\begin{aligned} W &= ((V_D)^T V_D + \alpha I)^{-1} (V_D)^T V_N \\ \hat{G} &= V_D W \\ \hat{V_N} &= V_N - \hat{G} \end{aligned}$$

Where $\hat{V_N}$ are the debiased vectors.

2.4 Word Embedding Fairness Evaluation (WEFE)

From the metrics described in Section 2.3.1 we see that each one of them has its own functioning and inputs. Word Embedding Fairness Evaluation (WEFE) is an open source library, developed at the Department of Computer Science of the University of Chile, that had as its initial objective the encapsulation, evaluation and comparison of fairness metrics [5].

To achieve its objective of standardizing how the metrics are executed, WEFE defines a common structure that is explained in the next section.

2.4.1 WEFE Bias Measurement Structure

WEFE implements all of the metrics described in section 2.3.1, except for Direct and Indirect bias. To standardize the bias measurement metrics a common usage pattern is defined; the main parts forming the bias measurement framework are presented in this section.

Target

A target set (T) is a set of words that can characterize a certain social group, which is defined by a certain criterion. This criterion can be any quality than distinguishes groups of people, like gender, age nationality, etc. For example, in the case of gender, two target sets can be defined one for "*man*" and one "*woman*". This way the target set representing "*woman*" can include words like "*she*", "*woman*","*girl*", etc. And the target set for "*man*" can include words like "*he*","*man*","*boy*", etc.

Attribute

An attribute set (A) is a set of words that represent some characteristic, trait, occupational field, etc. that can be associated to individuals of any social group. For example the attribute set for "*science*" can include words such as "*math*","*physics*","*chemistry*", etc. Meanwhile, the attributes set for "*art*" would include words such as "*poetry*","*dance*", "*literature*" etc.

Query

A query is defined as a pair $Q = (T, A)$, where T is a set of target words, and A is a set of attribute words. For example, having target word sets T_{women} and T_{men}:

$$T_{women} = \{she, woman, girl...\}$$
$$T_{men} = \{he, man, boy...\}$$

And the attribute word sets $A_{science}$ and A_{art}

$$A_{science} = \{math, physics, chemistry...\}$$
$$A_{art} = \{poetry, dance, literature...\}$$

This way, a query in WEFE, containing target words for "*man*" and "*woman*" and attributes sets "*science*" and "*art*", would be defined as follows:

$$Q = (\{T_{women}, T_{men}\}, \{A_{science}, A_{art}\})$$

Metrics in WEFE

Finally, a metric in WEFE is represented by a function that receives a query and a word embedding model as input and produces a real number as output.

For comparing word embedding models using multiples queries and multiple metrics a ranking process is defined, which is explained next.

Ranking process

When using multiple models, queries and metrics, a ranking process is proposed in WEFE in order to compare the results obtained.

Having a set of queries $\mathcal{Q} = \{Q_1, Q_2, ..., Q_r\}$, a set of models $\mathcal{M} = \{M_1, M_2, ..., M_n\}$ and a set of metrics $\mathcal{F} = \{F_1, F_2, ..., F_m\}$, $F_i(M_i, Q_i)$ is computed for every model in $\mathcal{M}$, every query in $\mathcal{Q}$ and every metric in $\mathcal{F}$. This way, for a fixed model and metric, a vector of scores is obtained where every component corresponds to the score of a different query. This is repeated for every model and metric.

With the scores obtained, a matrix is constructed for every metric, with dimensions $|\mathcal{M}| \times |\mathcal{Q}|$. To create the ranking it is necessary to aggregate the results by embedding model, done by simply calculating the means of the scores obtained. This way, only one value for every combination of model and metric.

Then, is constructed a final matrix with dimensions $|\mathcal{M}| \times |\mathcal{F}|$. Next, a rank is created indicating the most and least biased models according to the metrics.

2.4.2 WEFE Case Study

In the original WEFE paper [5] a case study is conducted instantiating the framework to compare six publicly available pre-trained word embedding models, using four of the metrics described in Section 2.3.1. As it was explained, the functions representing the metrics receive as input a word embedding model and a query. In this case study the models and the queries used are the following:

Models

Here we briefly describe the pre-trained word embedding models used in the case study. Most of these models were obtained using the Gensim library interface[3] and Lexvec is obtained from its original source[4] [4].

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

[4] https://github.com/alexandres/lexvec

1. Conceptnet [29]: Model trained over Conceptnet, word2vec, Glove, and OpenSubtitles 2016; its vectors have 300 dimensions.
2. Fasttext-Wikipedia [6]: Model trained over Wikipedia 2017, UMBC webbase corpus and statmt.org news; its vectors have 300 dimensions.
3. Glove-Twitter [26]: Model trained over 2B tweets; its vectors have 200 dimensions.
4. Glove-Wikipedia [26]: Model trained on Wikipedia 2014 + Gigaword 5; its vectors have 300 dimensions.
5. Lexveccommoncrawl [27]: Model trained on Common Crawl; its vectors have 300 dimensions.
6. Word2vec-Googlenews [24]: Model trained on Googlenews; its vectors have 300 dimensions.

Queries

In the case study 25 queries are used to measure bias for three different criteria. The criteria and the number of queries for each one are the following:

1. Gender (Q_{gender}): 7 queries
2. Ethnicity ($Q_{ethnicity}$): 9 queries
3. Religion ($Q_{religion}$): 9 queries

For each criteria, the ranking process described above is performed and a bias ranking is obtained for each of the criterion, where the higher the position the greater the bias.

Partial results of the WEFE case study are shown below.

Results

Since in this work we focus on gender bias, only the results of gender obtained by the case study are shown. In Table 2.1 all the scores and rankings for the models are exhibited, while in Figure 2.1 demonstrates the total ranking of gender bias for the embeddings model, showing the models from the most to the least biased.

2.4.3 Bias Mitigation

In Section 2.3.2 several bias mitigation algorithms were explained. Each of the algorithms performs different operations and takes different inputs, making their use quite distinct.

WEFE implements two bias mitigation algorithms: Hard Debias (HD) [7] and Multiclass Hard Debias (MHD) [22], generalization of HD to support more than two bias criterion.

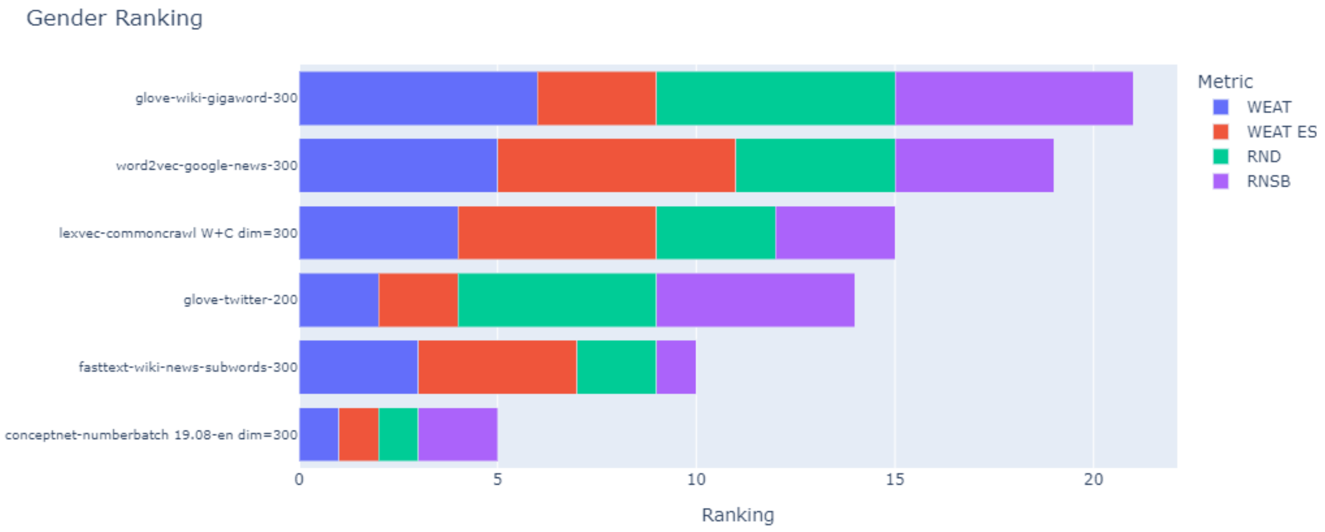

Figure 2.1: Original gender ranking in WEFE case study

Model Name	WEAT	WEAT ES	RND	RNSB
conceptnet-numberbatch 19.08-en dim=300	1 (0.202)	1 (0.366)	1 (0.007)	2 (0.02)
fasttext-wiki-news-subwords-300	3 (0.468)	4 (0.709)	2 (0.018)	1 (0.018)
glove-twitter-200	2 (0.411)	2 (0.504)	5 (0.127)	5 (0.049)
glove-wiki-gigaword-300	6 (0.845)	3 (0.656)	6 (0.183)	6 (0.074)
lexvec-commoncrawl W+C dim=300	4 (0.533)	5 (0.762)	3 (0.042)	3 (0.021)
word2vec-google-news-300	5 (0.83)	6 (0.941)	4 (0.084)	4 (0.033)

Table 2.1: Scores and rankings for all metrics and embedding models of the original WEFE case study

Nevertheless, to achieve the library goals it is necessary to incorporate more algorithms adjusting them to the standardization.

Chapter 3

Implemented Methods

In this chapter we present the core of the work done, including the study of the algorithms and their implementation in WEFE. First, in Section 3.1, we describe some similarities found among the algorithms that made possible their standardization. Then, in Section 3.2, we describe how the bias mitigation (also known as debias) is standardized and how we adapted the algorithms to fit into WEFE. Finally, in Section 3.3, we discuss the update done to the WEB framework in order to use it to evaluate the performance of the word embedding models.

3.1 Similarities between algorithms

Even though the bias mitigation algorithms present notorious differences in their functioning, it is possible to find some similarities between them, such as the word sets used in the process or certain operations performed to make the debias possible. In this Section we discuss the similarities found between the algorithms.

3.1.1 Gender Bias

All of the studied algorithms were intended to mitigate gender bias in the word embedding models, using gender related word sets to do so. We notice that the relation they have to gender is determined only by the word sets they use, so it is possible to use them for other criteria by leaving the word sets passed to the algorithms as a parameter to be specified by the user.

3.1.2 Operations

Reviewing the algorithms mentioned in Section 2.3.2 we noted that, in general, they need to calculate certain values that are used in the debias process. These calculations can be seen as pre-operations to the debias.

Double Hard Debias needs to calculate the embedding's mean, perform a PCA over the set of vectors and obtain the "bias subspace". Repulsion Attraction Neutralization and Hard Debias need, as well, the "bias subspace", being equivalent for all three algorithms. On the other hand, Half Sibling Regression needs to perform a ridge regression to obtain the bias information.

With the process described above, we can recognize that the debias process, in general, is done in two steps: The pre-calculation of needed information and the actual debias. This generalization is useful given the proposed objective of standardizing the bias mitigation methods.

Bias subspace

Between the pre-operations described previously finding bias subspace, or bias direction, proposed originally by Bolukbasi et al.[7], is a common operation among the bias mitigation algorithms, being present in three of them.

Bias direction $\vec{g} \in \mathbb{R}^d$ largely captures bias in the embedding. This direction helps us to quantify direct and indirect biases in words and association [7].

To identify the bias subspace, word pairs that capture both bias groups are used, for example, in the case of gender, "*male*"-"*female*", "*he*"-"*she*", "*girl*"-"*boy*". Next, the pairwise difference vector is taken and its principal component is calculated, where the first component would correspond to the bias subspace.

3.1.3 Word Sets

The debias process is performed over a subset of the vocabulary, because there are some words that we do not want to debias. This leads to the algorithms to define sets of words for different purposes.

In general, three word sets are identified by the algorithms:

- **Bias Definition**: This set consists in words that define the bias groups. The bias in these words is considered important for its definition; therefore the debias process it is not applied to them. An example of words included in this set for gender bias is: "*man*", "*woman*", "*mom*", "*dad*", "*beard*", "*bikini*", etc.
- **Neutral Bias Words**: This set contains words that should be neutral to the bias to be mitigated, hence the debias is applied to them. In most of the algorithms this set is defined as the complement of the Bias Definition set, these is all the words in the vocabulary not considered as bias definition. Examples of words included in this set for the gender bias case are: doctor, nurse, programmer, friend, etc.

- **Definitional Pairs**: This set is used to define the bias direction described above, consists in a set of word pairs that define the bias. For example, for gender: man-woman, he-she, girl-boy.

3.1.4 Comparison between the algorithms

As previously mentioned, the algorithms present similarities and differences between each other. In Table 3.1 we present a comparative table between the algorithms. In the table we notice all of the algorithms need a set of bias definition words to operate: two of them normalize the vectors of the models, three of them subtract bias from vectors in some form and two of them have a way to distance words from the bias direction, among other things.

This comparison is useful to find common operations between the algorithms and common inputs, making easier the standardization. The analysis is also important for understanding how the algorithms mitigate the bias in the word embedding models, in order to identify what operations are the ones that most affect bias.

		HD	DHD	HSR	RAN
Word sets used as input	Definitional Pairs	√	√	×	√
	Bias Definition	√	√	√	√
Operations	Normalization	√	×	×	√
	Bias Direction	√	√	×	√
	Subtract Bias	√	√	√	×
	Distances words from bias direction	√	×	×	√

Table 3.1: Comparative table between the algorithms

3.2 Bias Mitigation Standardization

In order to standardize the bias mitigation algorithms in WEFE, we took an object-oriented approach. For this we use a template method design pattern, intending to obtain a modular solution. This way the bias mitigation module is easily extensible for more algorithms to be implemented in the future.

To encapsulate the algorithms, WEFE has its own class “BaseDebias”, that inherits from the “BaseEstimator” class from Scikit-learn[1] [8]. “BaseDebias” represents bias mitigation algorithms in WEFE and inheritates the basic data transformation from Scikit-learn: the fit-transform interface.

The fit-transform interface consists in having all the debias process done by two methods, “fit” and “transform”. In the first step, “fit”, all the corresponding transformations needed to perform the debias are calculated; this consists of the pre-calculations discussed in Section

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.base.BaseEstimator.html?highlight=baseestimator#sklearn.base.BaseEstimator`

3.1.2. Then, the method "transform" applies the transformations learned in 'fit", to actually perform the debias process.

Every bias mitigation algorithm included in WEFE must be an instance of the class "BaseDebias", and thus must implement the fit-transform interface. This class hierarchy is shown in the class diagram we exhibit in Figure 3.1.

In Section 3.1.3 the word sets "Bias Definition" and "Neutral Bias Words" were described. In WEFE these sets are defined as ignore and target set respectively. These sets are passed as parameters to the algorithms in the transform method and define the words that must be ignored during the debias (ignore) and the words that must be debiased (target). The flow of the WEFE bias mitigation process is shown in Figure 3.2.

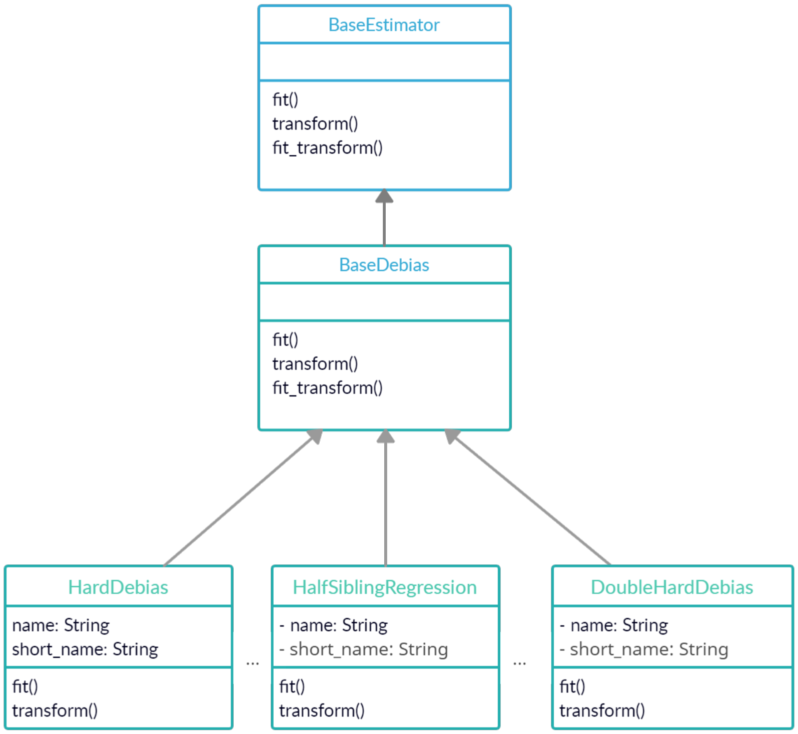

Figure 3.1: WEFE class diagram for bias mitigation

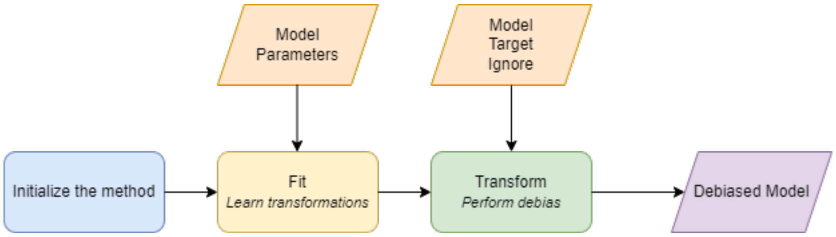

Figure 3.2: WEFE debias flow

Another important thing to note is that the algorithms have been generalized to all types of bias since they were originally intended to deal with gender bias. This is done by simply allowing the user to provide the sets of words necessary for the algorithm to work, thus being able to define the bias criterion they deem appropriate.

To implement the algorithms into WEFE they had to be adapted to the fit-transform interface already described. Next, we detail how each algorithm is adjusted to fit in WEFE.

3.2.1 Double Hard Debias

Double Hard Debias works with the assumption that word frequency in training corpora affects the bias of the model, so it removes the frequency direction that distracts the bias subspace. The adaptation of this algorithm to the fit-transform interface is done as the follows:

In the fit method, the bias subspace is obtained from the definitional pairs. Then, the mean of the entire set of vectors conforming the word embedding is calculated and a principal components analysis is applied to these vectors. The results of all of these calculations are saved for use during the debias process.

The transform method is performed in three steps:

1. The words to apply the debias are obtained by finding the ones that are most biased.
2. The components that reduce bias the most are searched by performing the debias once for each component obtained in the PCA; this is Hard Debias plus subtracting one of the components from the vectors, over the words obtained previously.
3. When the component is found the debias is applied to the words using the optimal component, obtaining new vectors that are updated in the model.

In order to incorporate target and ignore parameters to the algorithm, if ignore is specified, the words included in the set are simply not considered as candidates when searching for the most biased words. If the target is specified the process of searching the most biased words is applied to only that set of words.

3.2.2 Half Sibling Regression

Half Sibling Regression is a simple method that focuses on learning the bias information from the embeddings and then removing it to obtain vectors with no bias. The adaptation of this algorithm to the fit-transform interface is done in the following way:

First, in the fit method, a list of n bias definition words is passed as argument; these are words that are associated with bias by definition. With this a list of m non bias definition words is obtained, that correspond to the rest of the vocabulary not included in the bias definition words and the words to be debiased.

With both word lists defined, the word vectors are obtained to perform the operations, obtaining two matrices, $d \times n$ and $d \times m$, with d the dimension of the embedding. In these matrices each column represents a word vector; hence the number of columns is the numbers of words represented in the matrix and the number of rows corresponding to the dimension of the embedding.

Next, the weight matrix is computed, which is used to calculate the bias information that will be removed later. The bias information is an $m \times d$ matrix, where each column of the bias information matrix corresponds to a column in the matrix of words to be debiased.

Then, in the transform method, the bias information is subtracted from the non-bias definition words. Finally, the vectors are updated in the embedding model.

Transform receives as parameters target and ignores sets. If these sets are not specified the debias is applied to all non bias definition words. If these sets are included the debias is applied to the corresponding words.

To adapt the algorithm to be performed over a subset of the non bias words, if the target or ignore parameters are specified, then the indexes of the specified words in the matrix of words to be debiased are obtained; this way only the corresponding columns to the specified words are considered in the subtraction of the bias information.

An example of this would be, having a vocabulary: { *"man"*, *"woman"*, *"doctor"*, *"nurse"*, *"programmer"*} and a list of gender definition words: { *"man"*, *"woman"*}; then the matrices with the word vectors, V_n and V_d and the bias information G would be the ones in Figure 3.3.

When the debias is done to the entire set of non bias definition words (i.e { *"nurse"*, *"doctor"*, *"programmer"*}) simply $V_n - G$ is computed. When target words are specified, for example { *"nurse"*, *"programmer"*}, only the columns corresponding to theses words in V_n and G are considered; this is shown in Figure 3.4. Then, $V_n - G$ is computed to obtain the debiased vectors.

$$\text{Vn} = \begin{bmatrix} 2 & 1 & 5 \\ 3 & 4 & 7 \\ 5 & 1 & 3 \end{bmatrix} \qquad \text{Vd} = \begin{bmatrix} 0 & 1 \\ 2 & 0 \\ 1 & 2 \end{bmatrix}$$

$$\text{G} = \begin{bmatrix} 91 & 75 & 141 \\ 400 & 100 & 366 \\ 383 & 200 & 466 \end{bmatrix}$$

Figure 3.3: Examples of matrices used for Half Sibling Regression

3.2.3 Repulsion Attraction Neutralization

Repulsion Attraction Neutralization, creates a transformation that reduces the bias in the embeddings minimizing the semantic offset. The adjustment of this algorithm into WEFE is done such that the fit and transform method implement the following:

$$Vn = \begin{bmatrix} 2 & 5 \\ 3 & 7 \\ 5 & 3 \end{bmatrix}$$

$$G = \begin{bmatrix} 91 & 141 \\ 400 & 366 \\ 383 & 466 \end{bmatrix}$$

Figure 3.4: Examples of selected target vectors to perform the debias

In the fit method the bias subspace is obtained from the definitional pairs. This is then saved to use it when needed during the debias process.

In the transform method we perform the debias; this is done by iterating through the target words. During this process, first, a copy of the original vector is made to find its debiased counterpart that minimizes the objective function from Equation 2.2. With this, the debiased vectors are obtained and stored to update them in the model.

The adaptation of this algorithm to choose the words to debias is done by iterating only through the words passed in the target parameter if it is specified and skipping the ones included in the ignore parameter if specified. When these parameters are not specified the debias is done to all the words in the vocabulary.

3.2.4 Evaluation

In order to evaluate the new algorithms included in WEFE, we take an experimental approach by experimenting with the new methods.

In Section 4.1 we present a comparison between WEFE implementation of the algorithms and the original to check their correct implementation. In Section 4.2.1 we apply the algorithms to several word embeddings models to compare the bias mitigation among them, making sure that there is actually a reduction in the bias.

3.2.5 Development Process

During the process of integrating the algorithms into WEFE we adopted some good development practices in order to obtain good quality code, easy to read and maintain. These practices include continuous development, the use of branches and pull requests on Github and documenting all the new functionalities added.

Besides the described above, we implemented unit tests for every new functionality included in order make sure that everything works as is intended.

3.2.6 WEFE Repository

We implemented all of the above mentioned in WEFE's Github repository[2] and is for now available in the "develop" branch[3]. These new functionalities in WEFE will lead to a new release of the software in the near future.

3.2.7 WEFE Documentation

We include the documentation for all the new algorithms incorporated into WEFE in the WEFE documentation[4] and is for now available in the develop branch[5]. In here we include the API documentation and examples of how to use each of the algorithms.

3.3 Word Embeddings Benchmarks

As mentioned in Section 2.2.4, the Word Embeddings Benchmarks, has not been updated in over two years. Due to this lack of maintenance the software does not function properly.

To use this framework to evaluate the word embedding models, after applying the different bias mitigation process to them, it was necessary to update WEB to support new versions of the library it uses.

The changes done to the library were mainly updating the requirements to the current versions of the libraries WEB uses. This led to some changes in the imports found in WEB and the use of some methods from the libraries that were outdated.

Besides updating the libraries, a method meant for uploading Gensim word embedding models was fixed. This method had some issues that were resolved, therefore it could function properly; in addition to this support for Gensim version 4.0.0 was added given that it only supported versions below this one.

With the changes mentioned, the framework is now completely functional. We leave the updated version of the library available for public use in its own repository in the DCC Github[6].

[2] https://github.com/dccuchile/wefe

[3] https://github.com/dccuchile/wefe/tree/develop

[4] `https://wefe.readthedocs.io/en/latest/`

[5] `https://wefe.readthedocs.io/en/develop/`

[6] https://github.com/dccuchile/word-embeddings-benchmarks

Chapter 4

Experiments

This chapter presents all of the experiments conducted with the implemented algorithms, including, first, a comparison between the implemented algorithms and the original implementation, which allows to check their correct integration in WEFE. Then, a comparison between the different algorithms included in WEFE for six different word embedding models and performance of the debiased models using the updated version of the Word Embeddings Benchmark.

In Section 4.1 we show the results of comparing the implemented algorithms to their original versions. In Section 4.2.1 we show the comparison between the debiased models and in Section 4.2.2 we present the experiments done with WEB.

4.1 Correctness check

With the algorithms already implemented in WEFE it was necessary to check if the implementation impacts the bias in the word embedding models in the same way as the original, to make sure that the WEFE implementation is consistent with the original. To do this we apply the WEFE implementation of the algorithm to the word embeddings models used by the original implementations of the methods and compare the results. This is done only with two of the three methods: Half Sibling Regression and Double Hard Debias. For Repulsion Attraction Neutralization there were no models available to compare, which is why we will consider that the implementation is consistent with the original so long as it effectively reduces the bias in word embedding models.

4.1.1 Half Sibling Regression

In the original paper [28] this method is tested with a Glove model trained on a Wikipedia dump of English articles from 2017, using gender criterion for the debias. We download the original model and the debiased one and apply the WEFE implementation of Half Sibling Regression to the original model, then we measure the bias of each model using the gender

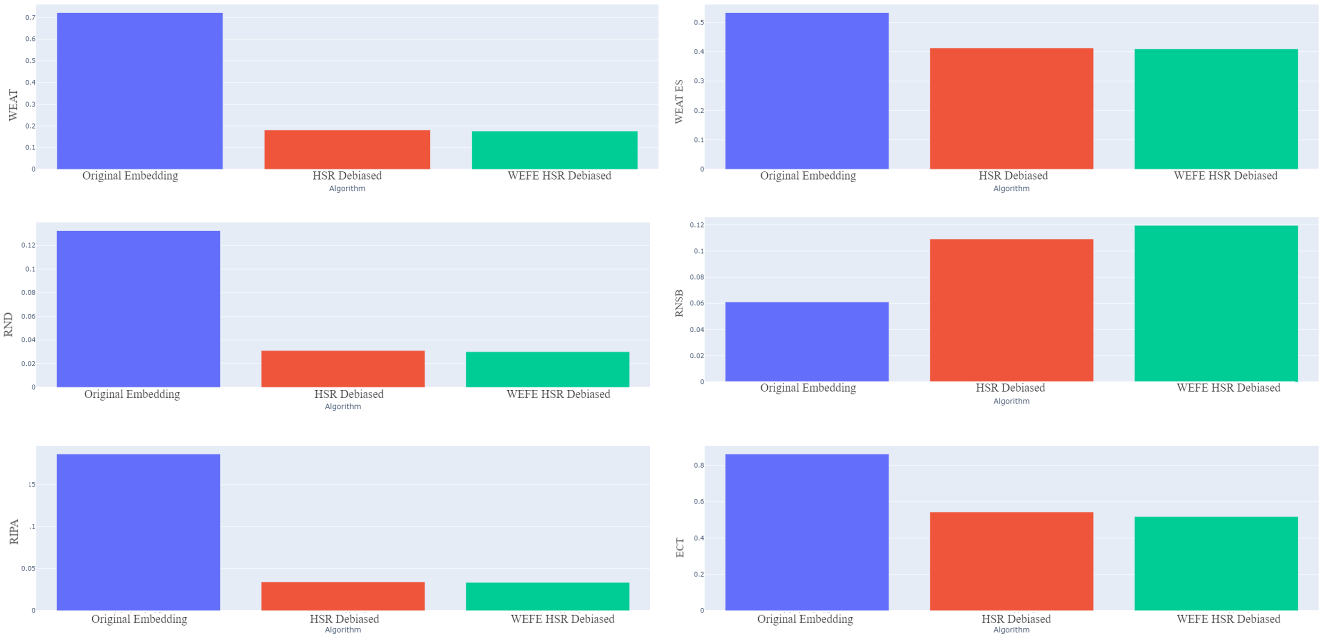

Figure 4.1: Comparison between word embedding model, HSR debiased and the debiased by the WEFE implementation of HSR

part of the WEFE case study. We show the results obtained from the comparison in Table 4.1 and Figure 4.1.

The results obtained from the comparison between both implementations show that both of them reduce the bias of the model almost the same. With these results we confirm that our implementation of Half Sibling Regression is consistent with the original.

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original Embedding	0.722	0.532	0.132	0.061	0.186	0.862
HSR Debiased	0.181	0.412	0.031	0.109	0.034	0.543
WEFE HSR Debiased	0.176	0.409	0.03	0.12	0.034	0.518

Table 4.1: Comparison between word embedding model, HSR debiased and the debiased by the WEFE implementation of HSR

4.1.2 Double Hard Debias

Originally Double Hard Debias [31] is tested on a Glove model. We download the original model, plus the debiased one from the data published by the creators and compare the original implementation to the WEFE implementation. As originally DHD is applied on gender criterion we also apply our version in gender criterion and compare them with the gender part of the WEFE case study.

During the process of comparing the models we noticed that the results in bias measurement between the WEFE implementation of Double Hard Debias and the original implemen-

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original Embedding	0.722	0.532	0.132	0.061	0.186	0.862
DHD Debiased	0.171	0.225	0.132	0.045	0.071	0.99
WEFE DHD Debiased	0.656	0.503	0.121	0.06	0.169	0.858
WEFE DHD Debiased including definitional pairs	0.344	0.29	0.113	0.053	0.119	0.934

Table 4.2: Comparison between word embedding model, DHD debiased and the debiased by the WEFE implementation of DHD

tation were very different. We then found that the difference between the result was caused because in the original implementation the definitional pairs are included in the list of words to be debiased.

Given that the definitional pairs are words that capture the bias of each group and the bias contained in these words is considered to carry semantic information. For the experiments presented in this work we do not include the definitional pairs in the words that are debiased, even though these results are presented in the Annexed A.

Table 4.2 presents the results of the comparison between the WEFE implementation of DHD and the original, including both models: Including the definitional pairs in the debias and not including them for the WEFE implementation.

These results reveal that the inclusion of the definitionals pairs impacts in how the bias is reduced by the method. It is also noted that both DHD implementations, Original and WEFE's, reduce bias in a similar way with some minor differences that could be attributed to some differences in the application of the PCA and will lead to further analysis in the future. Despite the small differences between both implementations we consider WEFE's implementation of Double Hard Debias is consistent with the original, due to the changes in the bias generated by both of them are close to the same.

4.2 Performance of Algorithms and Models

Once the algorithms are implemented, two major concerns about them arise. First, how effective are the debias methods? Secondly, if applying these methods to the word embedding models impacts in their performance representing the vocabulary.

To address both concerns mentioned above, we designed two experiments. First, the WEFE case study, explained in Section 2.4.2 is replicated using the same words embeddings models applying the bias mitigation algorithms to them; all of this is explained in Section 4.2.1. Then, in Section 4.2.2 we test the same models using WEB, in order to see if their performance is affected by the debias.

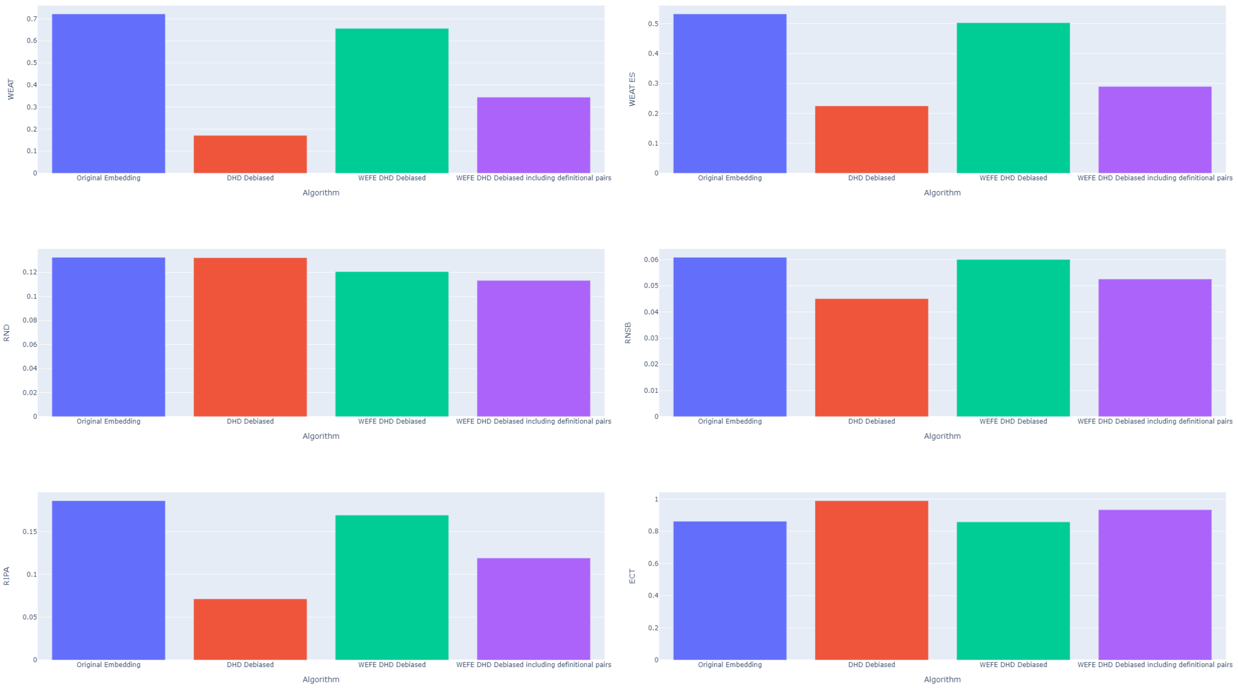

Figure 4.2: Comparison between word embedding model, DHD debiased and the debiased by the WEFE implementation of DHD

4.2.1 Bias in word embedding models

We use the WEFE case study, explained in Section 2.4.2, to measure how the bias changes after applying the different bias mitigation algorithms that WEFE offers and compare them to each other.

We do this by applying the algorithms available in WEFE: Hard Debias, Double Hard Debias, Half Sibling Regression and Repulsion Attraction Neutralization, to the six word embedding models used in the WEFE case study. Then, for each word embedding model we replicate the case study to compare the original model to the debiased versions.

The debias we apply to the models consists in gender debias, mainly for two reasons: first, it is widely studied and there are word sets available to work with, unlike other type of biases that are not as studied. Secondly, it is not a multiclass bias such as racial or religious which are incompatible with the algorithms. Therefore, we only use the gender queries of the case study, thus studying only the gender bias in the models.

When applying the algorithms we use the hyper parameters and configurations recommended by the original papers. Regarding the word sets used in the debias process, these can be found in Annexed B and consist in sets already included in WEFE. The sets are the “definitional pairs” used to find the bias subspace and the “gender specific” set used as the words that are excluded from the debias process: the ones included in the ignore parameter of the method transform.

In addition to the original model and the debiased ones, we include in the comparison a normalized version of the original model; this is the original model with all of its vectors normalized. This is because two of the algorithms, HD and RAN, include a normalization of the model in their debias process, and we therefore wish to research if the normalization of the model has any effect in the bias of the model.

Next, in Section 4.2.1 we present the results of applying the WEFE case study to one of the models, their normalized version and the debiased versions. The results for the rest of the models can be found in Annexed A.

Results

In this section we present the results obtained when applying the WEFE case study to compare the different implemented algorithms to the Glove-Twitter word embedding model. In Annexed A.1 we present the results obtained for the remaining word embedding models.

In Figure 4.3 we show a bias ranking for the different versions of the Glove-Twitter model, including the original, the normalized version and the debiased ones. This ranking shows how all the debiased models present less overall bias than the original one, as well as the normalized version.

Table 4.3 shows the results obtained from the metrics applied to the models, as well as the position of the models in the ranking for each metric. Then, Figures 4.4,4.5,4.6, 4.7,4.8 and 4.9 show bar charts for each metric comparing the models.

The results we present in this section and also the ones in the Annexed A, show that the bias mitigation algorithms that WEFE implements are capable of reducing the bias present in the model according to the metrics that WEFE offers. It is possible to notice in the gender rankings of Figure 4.3 and the ones presented in Annexed A.1.1, that the debiased versions of the models are positioned lower in the rankings, indicating that they are less biased.

The bar charts show how the bias changes for each metric among the models; in general the algorithms do reduce the bias, except in some cases that HSR increases it. HD and RAN are the ones that are more effective when it comes to decreasing the bias present in the models according to the metrics.

Double Hard Debias seems to not have much effect in the bias of the model, according to the results shown in this section (as confirmed by the results in Annexed A.1). But, when looking at the results in Table 4.4, which shows the results of applying Double Hard Debias to the model, but including the definitional pairs in the debias process, we notice that in this case the bias reduction is greater than when not including the definitional pairs (this can also be confirmed by the results in Annexed A.3).

Another interesting result obtained from the experiments is that the normalized models show less bias than the original ones, indicating that the simple act of normalizing the vectors has an effect in the bias measurement metrics. This opens up the question of why the normalization of the vectors affects bias and why it has different effects on the different metrics.

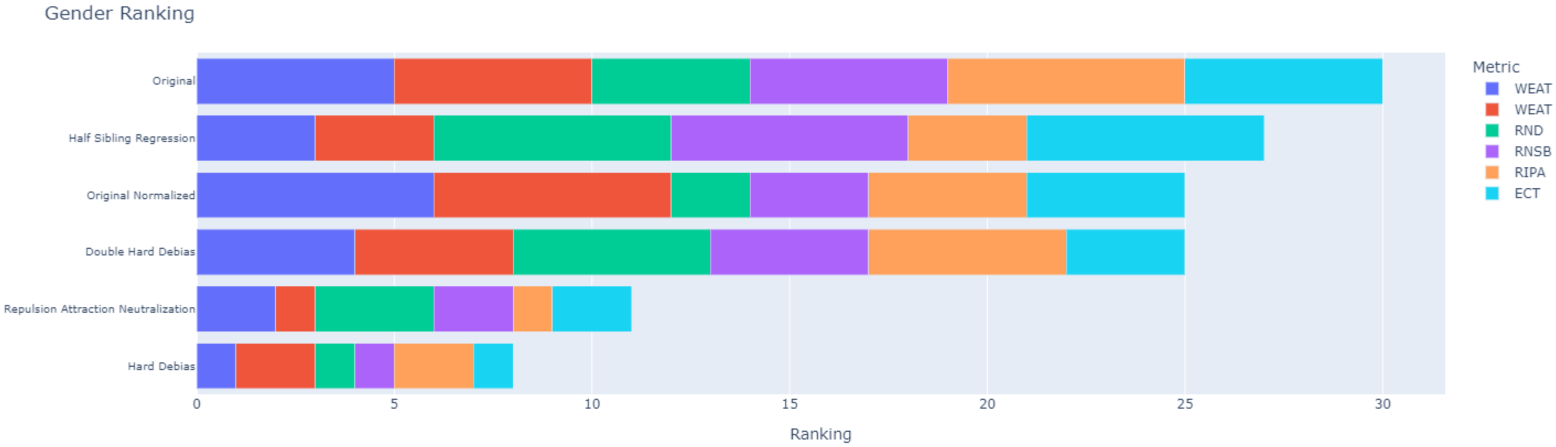

Figure 4.3: Ranking showing the most biased models for Glove-Twitter

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.411)	5 (0.504)	4 (0.127)	5 (0.05)	6 (0.211)	5 (0.942)
Original Normalized	6 (0.411)	6 (0.504)	2 (0.02)	3 (0.013)	4 (0.033)	4 (0.942)
Double Hard Debias	4 (0.332)	4 (0.44)	5 (0.131)	4 (0.048)	5 (0.19)	3 (0.943)
Half Sibling Regression	3 (0.133)	3 (0.436)	6 (0.262)	6 (0.122)	3 (0.033)	6 (0.474)
Repulsion Attraction Neutralization	2 (0.103)	1 (0.156)	3 (0.035)	2 (0.012)	1 (0.007)	2 (0.977)
Hard Debias	1 (0.084)	2 (0.174)	1 (0.006)	1 (0.012)	2 (0.008)	1 (0.992)

Table 4.3: Bias metrics results and rankings for Glove-Twitter

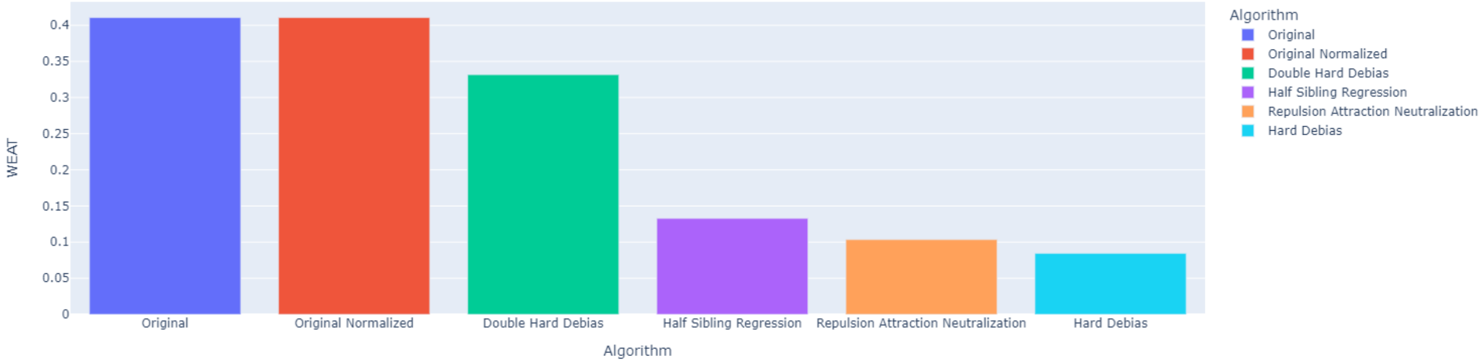

Figure 4.4: WEAT results for Glove-Twitter, closer to 0 is better

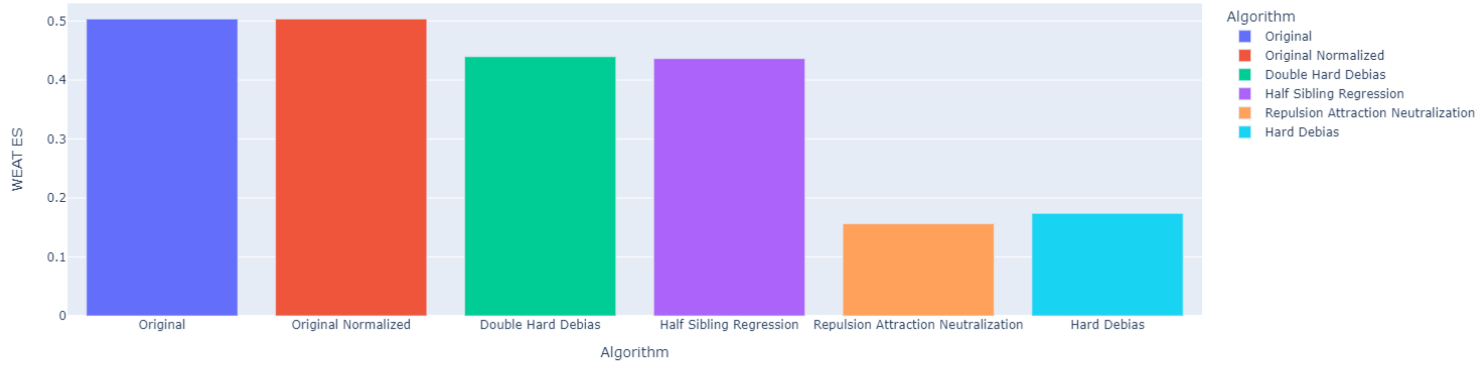

Figure 4.5: WEAT ES results for Glove-Twitter, closer to 0 is better

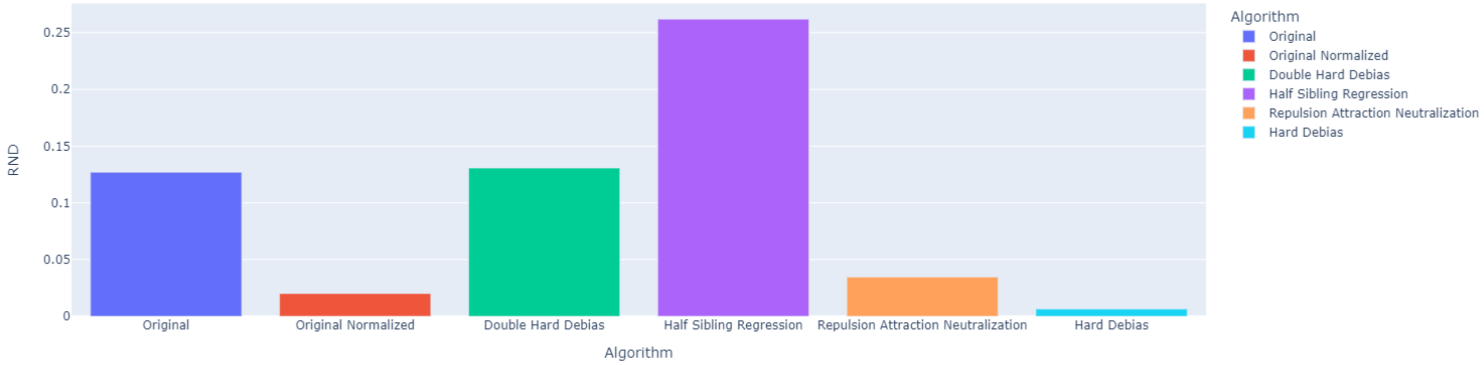

Figure 4.6: RND results for Glove-Twitter, closer to 0 is better

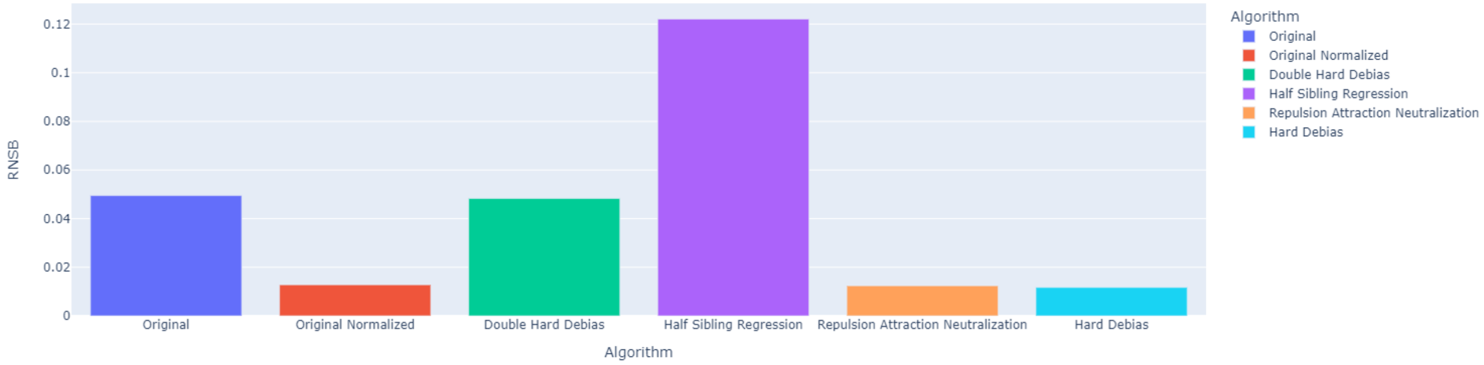

Figure 4.7: RNSB results for Glove-Twitter, closer to 0 is better

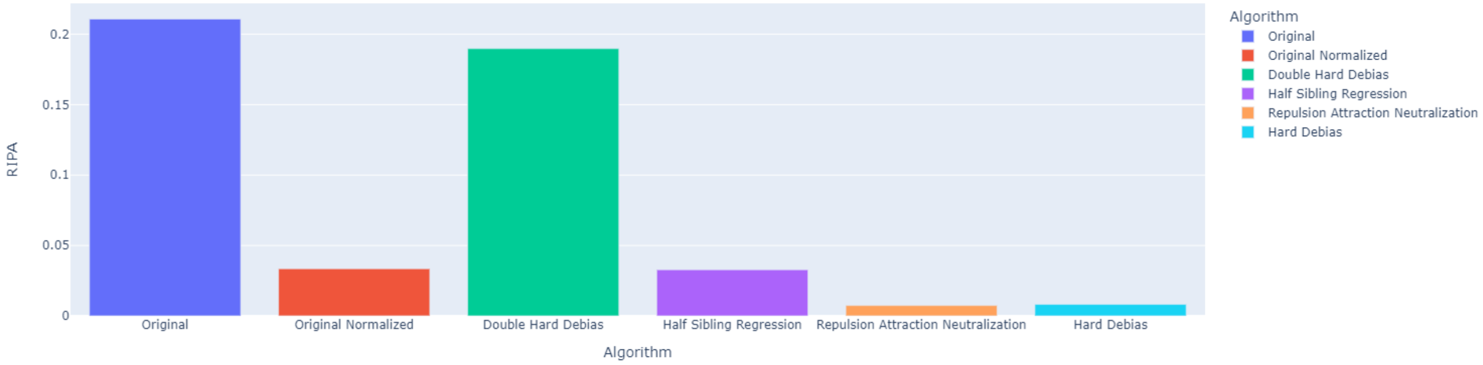

Figure 4.8: RIPA results for Glove-Twitter, closer to 0 is better

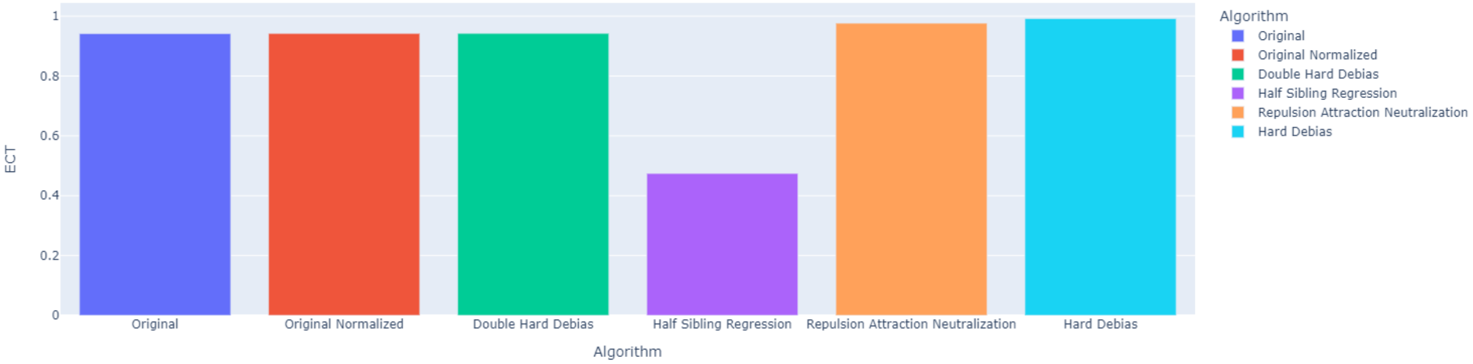

Figure 4.9: ECT results for Glove-Twitter, closer to 1 is better

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.411)	5 (0.504)	5 (0.127)	5 (0.049)	6 (0.211)	5 (0.942)
Original Normalized	6 (0.411)	6 (0.504)	2 (0.02)	3 (0.013)	4 (0.033)	4 (0.942)
Double Hard Debias	4 (0.222)	3 (0.294)	4 (0.05)	4 (0.044)	5 (0.16)	3 (0.96)
Half Sibling Regression	3 (0.133)	4 (0.436)	6 (0.262)	6 (0.128)	3 (0.033)	6 (0.474)
Repulsion Attraction Neutralization	2 (0.103)	1 (0.156)	3 (0.035)	2 (0.013)	1 (0.007)	2 (0.977)
Hard Debias	1 (0.084)	2 (0.174)	1 (0.006)	1 (0.012)	2 (0.008)	1 (0.992)

Table 4.4: Bias metrics results and Rankings for Glove-Twitter including the definitional pairs in the debias process when executing DHD

4.2.2 Performance of the word embedding models

Another important consideration to have when debiasing word embeddings models is not to alter the vectors in a way that affects the semantic information they contain, decreasing their performance in the different tasks they can solve.

The original papers of the methods claim that their debiasing process does not alter the functioning of the embeddings, and some even claim to improve their performance. In order to verify that this is actually true and that the models keep on being as functional as the original model after the debias, we use the updated version of the Word Embedding Benchmark framework (presented in Section 3.3) to compare the performance of the models before and after the debiasing process in three different tasks: analogy, categorization and similarity.

The comparison is done by taking each one of the original models used in the WEFE case study, their debiased version and the normalized version, and measuring their performance using all of the available datasets in WEB for the three tasks mentioned before. For each dataset a score is obtained. To have only one score per task, the simple mean of all the scores for each task is calculated.

The results of the recently explained experiments are presented below.

Results

In this section we show the most important results of measuring the performance of the word embeddings models in three different tasks using the WEB framework. The rest of the results can be found in Annexed A.

In general the results show that for the tasks "categorization" and "similarity" the debias and the normalization of the vectors does not alter the performance of the models. With "Analogy" it is possible to observe important changes for some models.

In Figure 4.11 we can notice that for the model Conceptnet there are no changes in the performance in the three tasks. On the other hand, in Figures 4.10 and 4.12 we can see that there are some notorious changes in "Analogy" for the models Glove-Wikipedia and Word2vec-Googlenews.

The major changes observed in the performances of the models is that the normalization of the vectors has an impact on how well a model performs in the task "Analogy"". This can be seen in Figures 4.10 and 4.12 where the normalized models and the debiased by RAN and HD (both methods normalize the vectors as part of the process) show improvement in the mentioned task.

Another thing to notice is that in some cases, as with the Glove-Wikipedia model in Figure 4.10, HSR worsens a bit of the performance of the model in "Analogy".

Finally, from the experiments presented in this section we can concluded that, in general, the debias does not have a negative effect on the performance of the word embedding models and that the normalization of the vectors of a model can greatly improve the performance of the model in "Analogy" tasks.

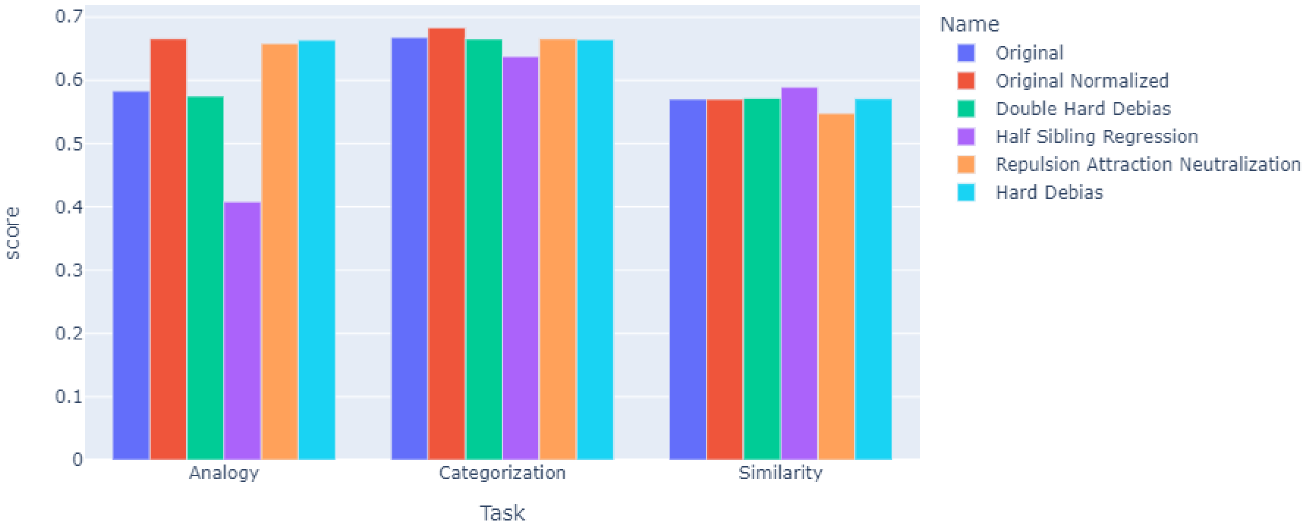

Figure 4.10: Glove-Wikipedia WEB Results

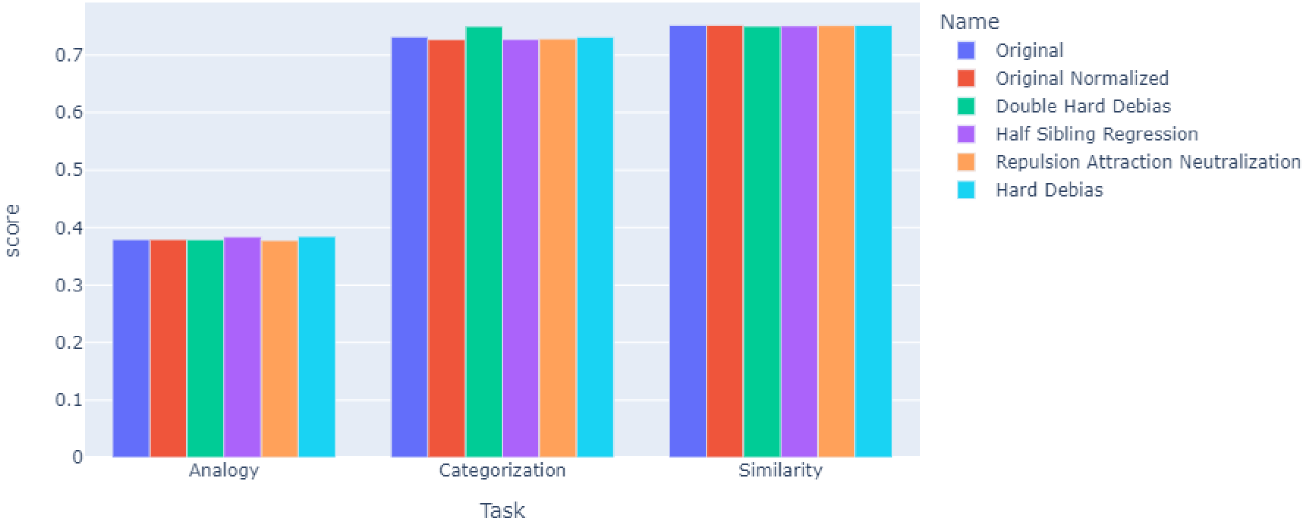

Figure 4.11: Concepnet WEB results

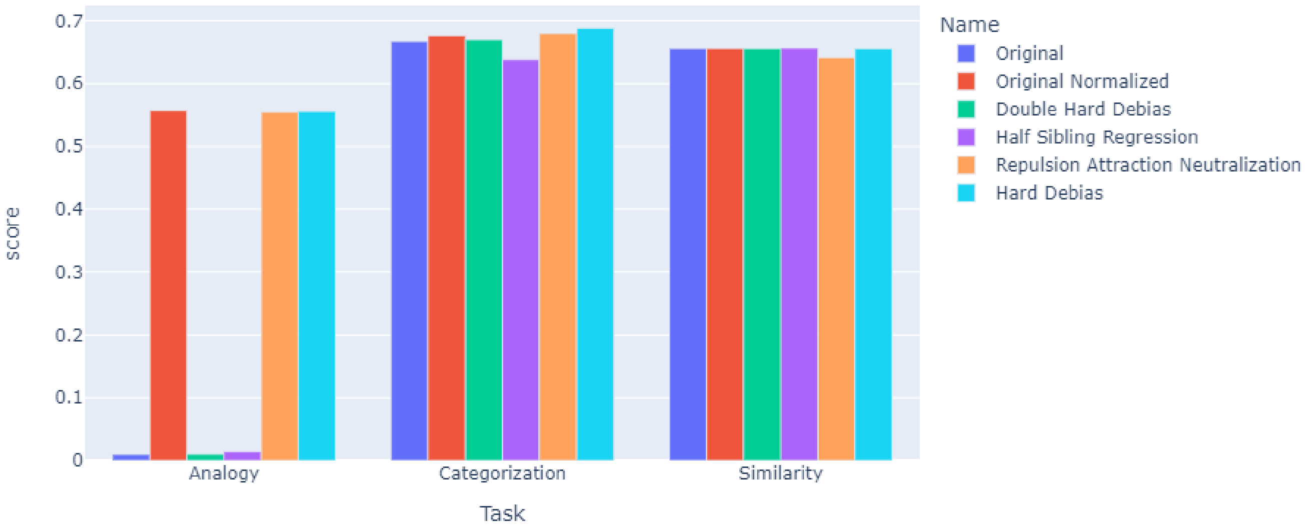

Figure 4.12: Word2vec-Googlenews WEB results

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this report we presented how three bias mitigation algorithms were standardized and included in WEFE. We also conducted several experiments applying the algorithms in order to check their correct implementation, compare them between each other and study how they affect the performance of the word embedding models. Next we discuss how the work done is connected to the objectives specified in Section 1.3.1, and the conclusions obtained from the performed experiments.

The first two objectives: *Design a method to unify the algorithms to be implemented so they can be used in a standard way, like the metrics already included in WEFE* and *Study different word embeddings bias mitigation approaches to fully comprehend their functioning so the integration into the framework is done properly.* The objectives were achieved by finding similarities between the algorithm and noticing they can fit into the fit-transform interface, standardizing the way of using them. This led to making WEFE a more complete library by incorporating more bias mitigation algorithms.

By updating the WEB library the third objective was achieved: *Continue the work done in the Word Embeddings Benchmark to obtain a functional version of the software and to have a valid tool to evaluate the performance of word embeddings in classical tasks.*. The achievement of this objective resulted in a functional version of WEB available to the community.

Regarding the fourth objective: *Compare the performance of different word embedding models after applying bias mitigation algorithms to them, to make sure their functioning is not affected by the debias.* in Section 4.2.2 and Appendix A.2 we showed the results of comparing the debiased versions of the models to the original version and the normalized version. The results showed that the bias mitigation process has no negative effect on the performance of the models when performing the tasks "Analogy", "Similarity" and "Categorization". An interesting finding made with these experiments is that the normalization of the vectors in a word embedding models has an impact on their performance in the tasks already mentioned.

Finally, in Section 4.2.1 and Appendix A.1, we presented the results of comparing how the algorithms reduce bias in different word embedding models achieving the last of the proposed objectives: *Compare the effectiveness of the different bias mitigation algorithm implemented during this work using the metrics included in WEFE.* The results obtained showed that in general the algorithms do reduce the bias in the models, varying the magnitude of the reduction depending on the metric and the model. The results of these experiments also revealed that normalizing the vectors of a model has an impact on the results, in this case the bias of the model.

This way, by including the algorithms into WEFE we achieved all of the objectives, resulting in a new version of the library to be released soon.

5.2 Future Work

The implementation of the bias mitigation algorithms and the results obtained from the experiments open the door to further research in this field. First, by expanding even more the algorithms included in WEFE by implementing the Gender Preserving debias method [20], which represents an even greater challenge to standardize than the already implemented methods.

The impact that the normalization of the vectors has on the performance of the models in the different tasks measured by WEB and on the bias measurement metrics is something of great interest and it is worth studying the effect that this simple operation has on the models.

In this work we focused on mitigating and studying only gender bias. We do not know if the bias mitigation applied has an impact on other bias criteria as ethnicity or religion. Studying how different bias criteria relate to each other is another interesting research, because mitigating gender bias in a word embedding model could affect, for example, racial bias in the model.

It is also of great interest to delve deeper into which operations performed by the algorithms are the ones that have the most debias effect and how they could interact with each other. This could lead to mixing the operations that most affect bias to create a new method of bias mitigation that incorporates procedures proposed by different algorithms.

Finally, it would be of great interest to extend the work to other languages like Spanish, since all of the bias related has been done in English. This would require a study of the words that define bias groups.

Bibliography

[1] Wefe pypi stats. https://pypistats.org/packages/wefe. Accessed Aug. 16, 2022.

[2] Amazon scrapped 'sexist AI' tool. *BBC News*, October 2018. "Accessed Dec. 21, 2021".

[3] In 2016, Microsoft's Racist Chatbot Revealed the Dangers of Online Conversation. https://spectrum.ieee.org/in-2016-microsofts-racist-chatbot-revealed-the-dangers-of-online-conversation, November 2019. Accessed Dec. 21, 2021.

[4] Pablo Badilla. Wefe: the word embeddings fairness evaluation framework. Master's thesis, University of Chile, 2020.

[5] Pablo Badilla, Felipe Bravo-Marquez, and Jorge Pérez. Wefe: The word embeddings fairness evaluation framework. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 430–436. International Joint Conferences on Artificial Intelligence Organization, 2020.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[7] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR*, abs/1607.06520, 2016.

[8] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[9] Sunipa Dev, Tao Li, Jeff M Phillips, and Vivek Srikumar. On measuring and mitigating biased inferences of word embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7659–7666, 2020.

[10] Oxford Dictionary. Bias definition. https://www.oxfordlearnersdictionaries.com/definition/english/bias_1#:~:text=%2F%CB%88ba%C9%AA%C9%99s%2F-,%2F%CB%88ba%C9%AA%C9%99s%2F,not%20based%20on%20fair%20judgement. Accessed Jul. 18, 2022.

[11] Jacob Eisenstein. Natural language processing. `https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf`, 2018.

[12] Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. *CoRR*, abs/1810.04882, 2018.

[13] Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Understanding undesirable word embedding associations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1696–1705, Florence, Italy, July 2019. Association for Computational Linguistics.

[14] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.

[15] Yoav Goldberg. *Neural Network Methods for Natural Language Processing*, volume 37 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool, San Rafael, CA, 2017.

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

[17] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[18] Aylin Caliskan Islam, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora necessarily contain human biases. *CoRR*, abs/1608.07187, 2016.

[19] Stanislaw Jastrzebski, Damian Lesniak, and Wojciech Marian Czarnecki. How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *CoRR*, abs/1702.02170, 2017.

[20] Masahiro Kaneko and Danushka Bollegala. Gender-preserving debiasing for pre-trained word embeddings. *CoRR*, abs/1906.00742, 2019.

[21] Vaibhav Kumar, Tenzin Singhay Bhotia, and Tanmoy Chakraborty. Nurse is closer to woman than surgeon? mitigating gender-biased proximities in word embeddings. *CoRR*, abs/2006.01938, 2020.

[22] Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047*, 2019.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[24] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[25] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *CoRR*, abs/1702.01417, 2017.

[26] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[27] Alexandre Salle, Aline Villavicencio, and Marco Idiart. Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 419–424, Berlin, Germany, August 2016. Association for Computational Linguistics.

[28] Bernhard Schölkopf, David W. Hogg, Dun Wang, Daniel Foreman-Mackey, Dominik Janzing, Carl-Johann Simon-Gabriel, and Jonas Peters. Modeling confounding by half-sibling regression. *Proceedings of the National Academy of Sciences*, 113(27):7391–7398, 2016.

[29] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975, 2016.

[30] Chris Sweeney and Maryam 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*, pages 1662–1667, Florence, Italy, July 2019. Association for Computational Linguistics.

[31] Tianlu Wang, Xi Victoria Lin, Nazneen Fatema Rajani, Bryan McCann, Vicente Ordonez, and Caiming Xiong. Double-hard debias: Tailoring word embeddings for gender bias mitigation. *CoRR*, abs/2005.00965, 2020.

[32] Zekun Yang and Juan Feng. A causal inference method for reducing gender bias in word embedding relations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9434–9441, 2020.

Annexes

Annexed A

Experiments Results

The appendix features all the results to the experiments that were not shown in Section 4.

A.1 Bias in Word Embedding Models Results

A.1.1 Bias Rankings

This section shows the bias rankings for the embeddings models.

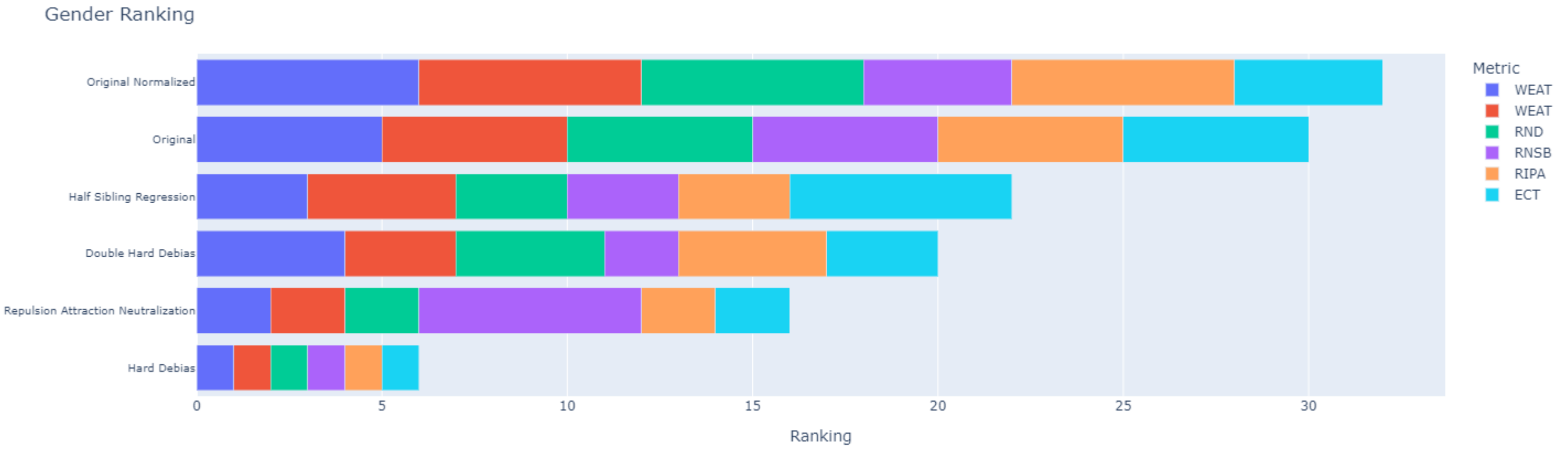

Figure A.1: Ranking showing the most biased models for Concepnet

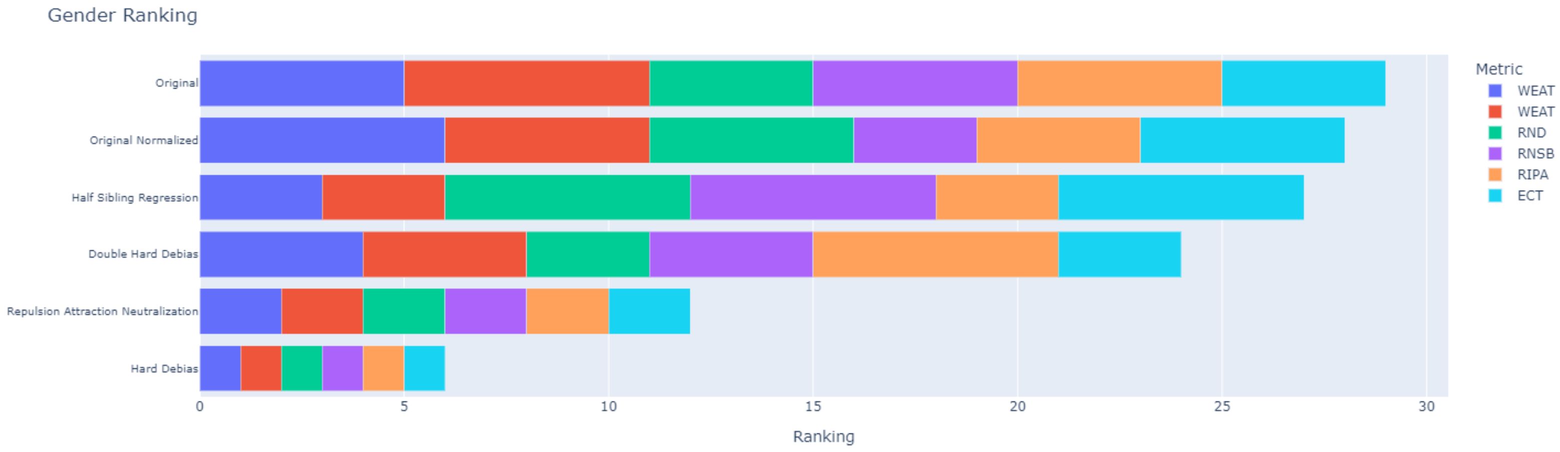

Figure A.2: Ranking showing the most biased models for Fasttext Wikipedia

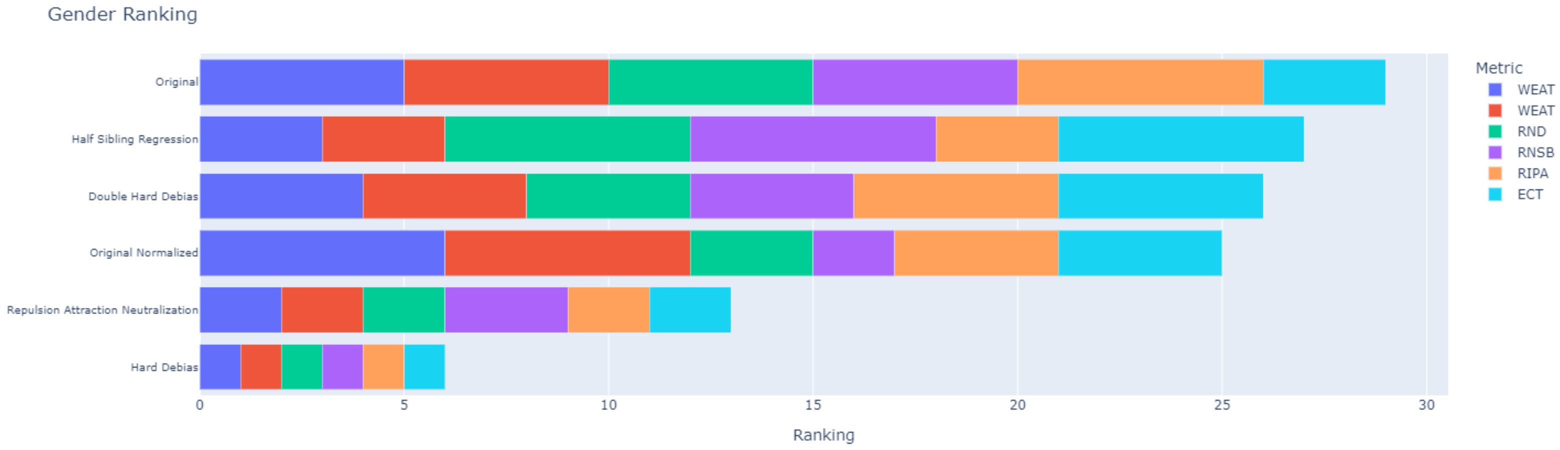

Figure A.3: Ranking showing the most biased models for Glove-Wikipedia

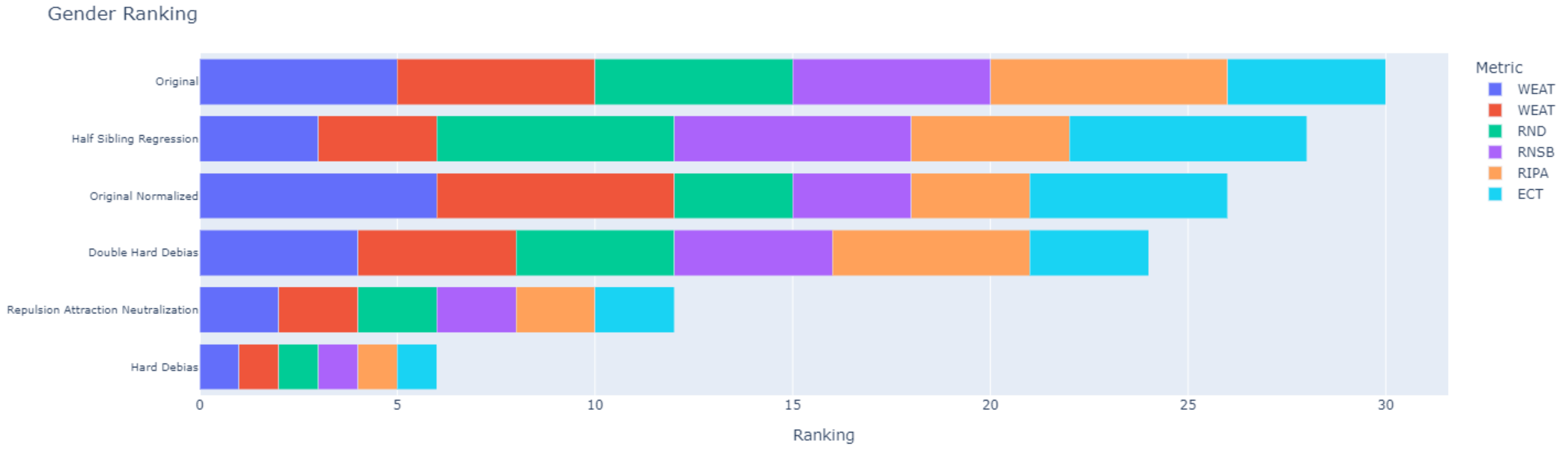

Figure A.4: Ranking showing the most biased models for Lexveccommoncrawl

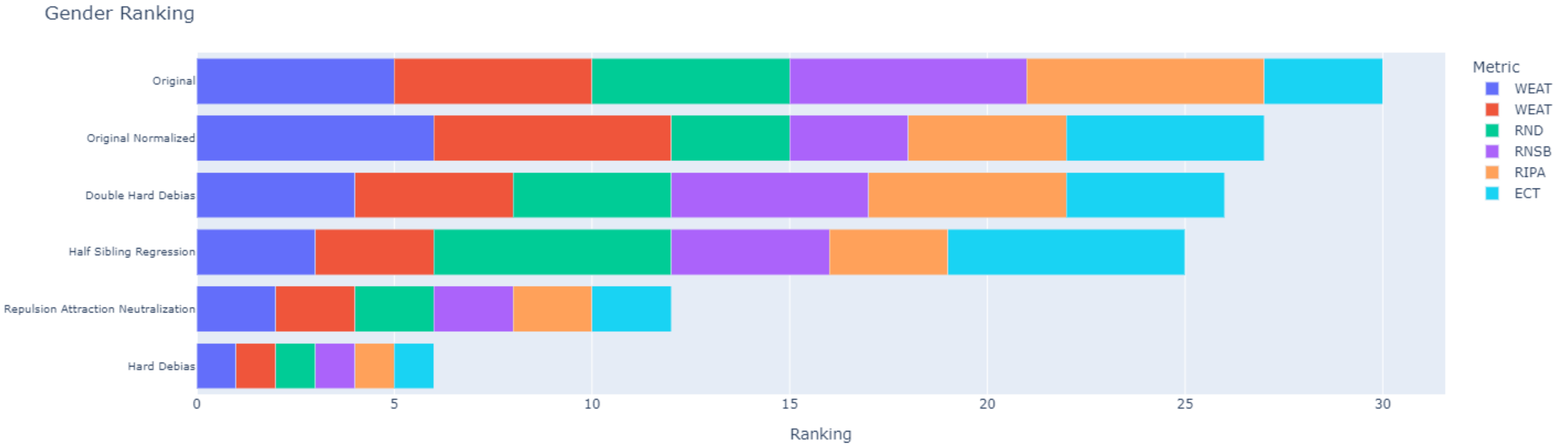

Figure A.5: Ranking showing the most biased models for Word2Vec-Googlenews

A.1.2 Bias Metrics

In this section we present the results obtained from the experiments, showing tables with the values for each metric, including the position of the model in the ranking. Bar charts are also included that compare the models for each metric.

Concepnet

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.202)	5 (0.366)	5 (0.007)	5 (0.02)	5 (0.01)	5 (0.913)
Original Normalized	6 (0.202)	6 (0.366)	6 (0.007)	4 (0.02)	6 (0.01)	4 (0.913)
Double Hard Debias	4 (0.192)	3 (0.361)	4 (0.007)	2 (0.019)	4 (0.01)	3 (0.914)
Half Sibling Regression	3 (0.173)	4 (0.364)	3 (0.006)	3 (0.019)	3 (0.008)	6 (0.862)
Repulsion Attraction Neutralization	2 (0.121)	2 (0.228)	2 (0.004)	6 (0.021)	2 (0.006)	2 (0.915)
Hard Debias	1 (0.045)	1 (0.092)	1 (0.002)	1 (0.018)	1 (0.002)	1 (0.987)

Table A.1: Bias metrics results and rankings for Conceptnet

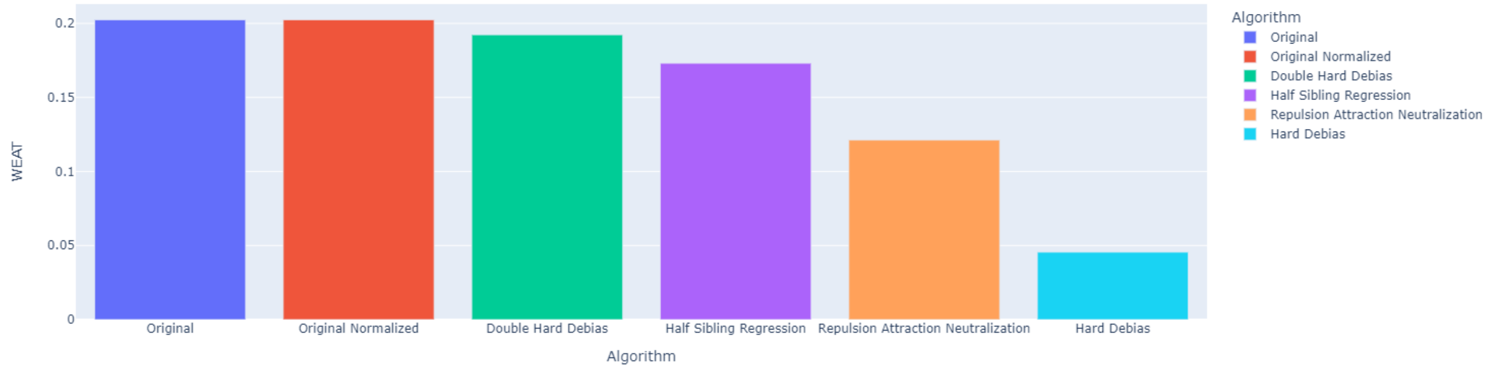

Figure A.6: WEAT results for Conceptnet

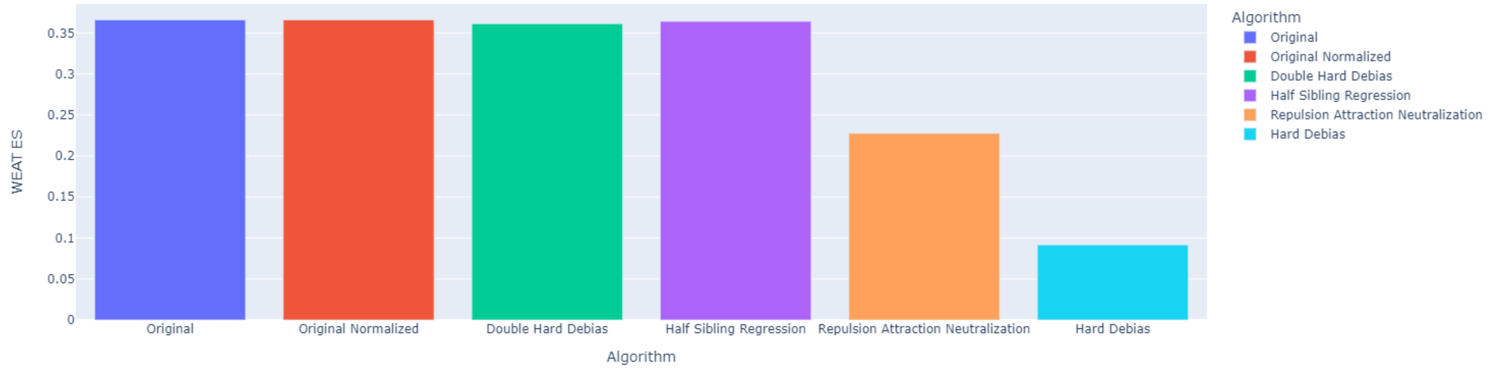

Figure A.7: WEAT ES results for Conceptnet

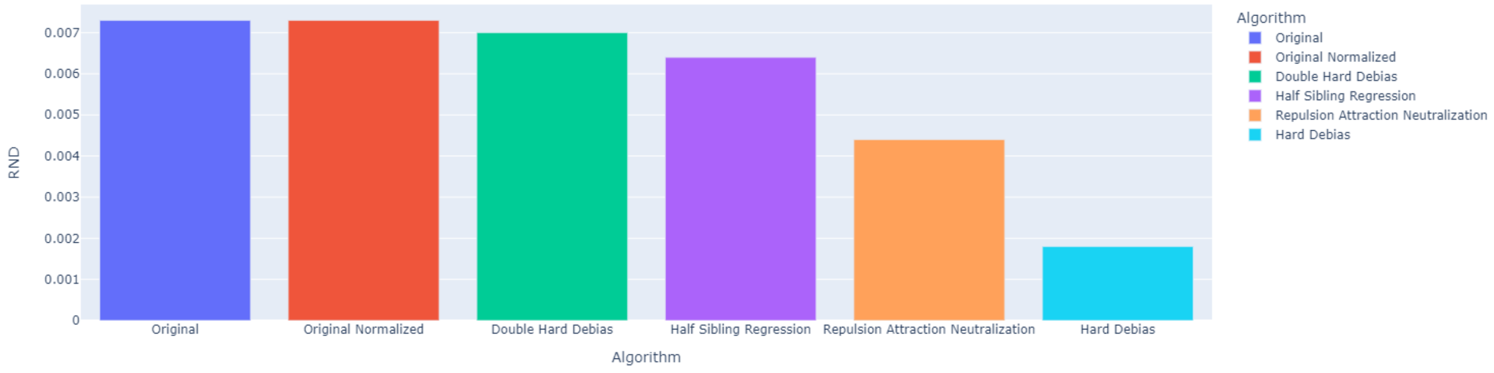

Figure A.8: RND results for Conceptnet

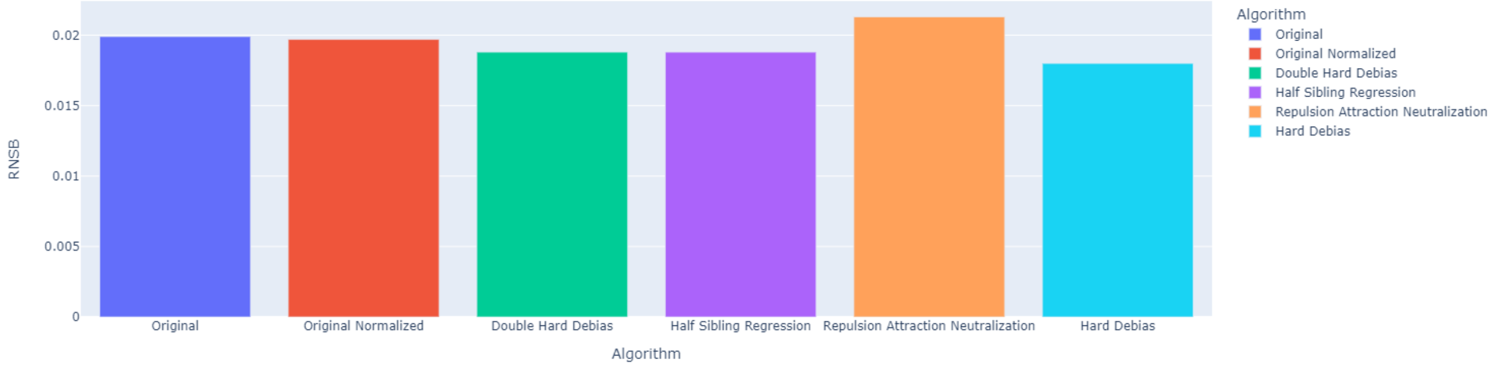

Figure A.9: RNSB results for Conceptnet

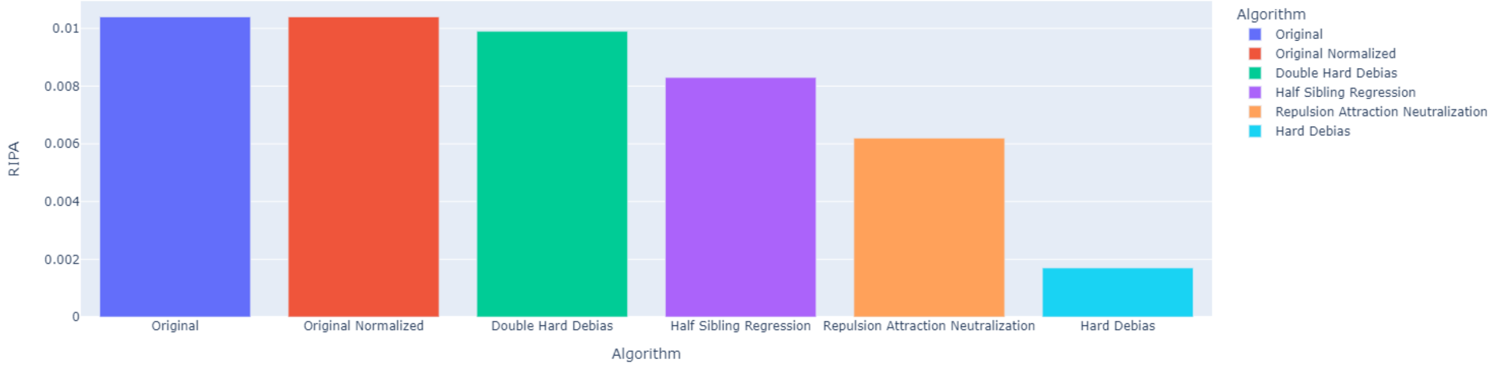

Figure A.10: RIPA results for Conceptnet

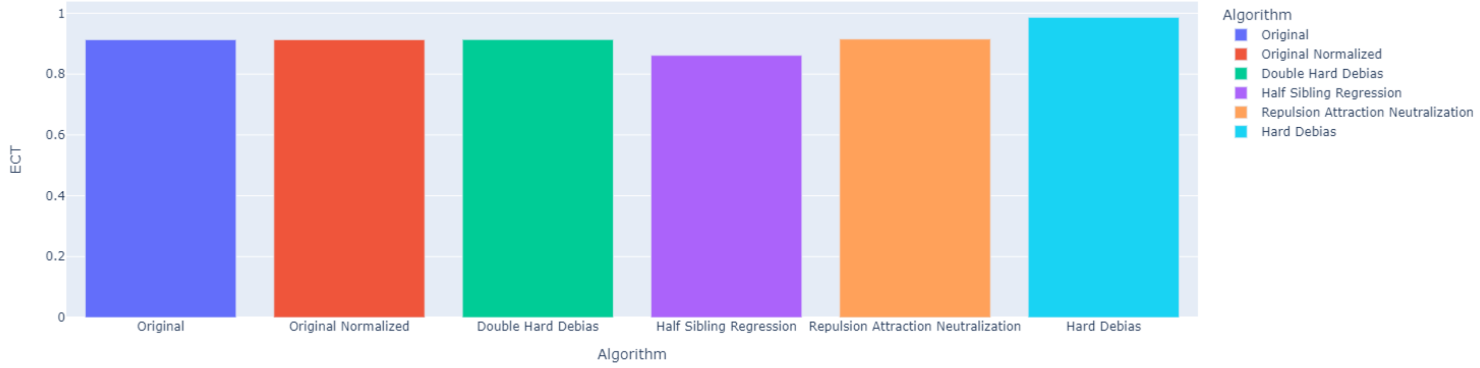

Figure A.11: ECT results for Conceptnet

Fasttext Wikipedia

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.468)	6 (0.709)	4 (0.018)	5 (0.018)	5 (0.05)	4 (0.888)
Original Normalized	6 (0.468)	5 (0.709)	5 (0.018)	3 (0.015)	4 (0.026)	5 (0.87)
Double Hard Debias	4 (0.437)	4 (0.694)	3 (0.017)	4 (0.017)	6 (0.051)	3 (0.9)
Half Sibling Regression	3 (0.436)	3 (0.676)	6 (0.029)	6 (0.018)	3 (0.023)	6 (0.762)
Repulsion Attraction Neutralization	2 (0.113)	2 (0.283)	2 (0.008)	2 (0.012)	2 (0.008)	2 (0.951)
Hard Debias	1 (0.055)	1 (0.148)	1 (0.003)	1 (0.011)	1 (0.003)	1 (0.984)

Table A.2: Bias metrics results and rankings for Fasttext Wikipedia

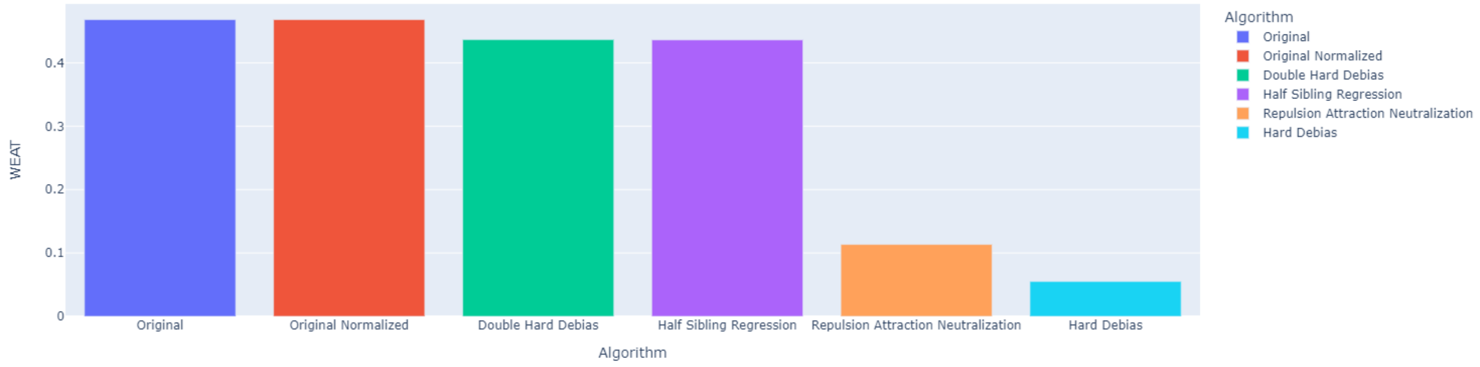

Figure A.12: WEAT results for Fasttext Wikipedia

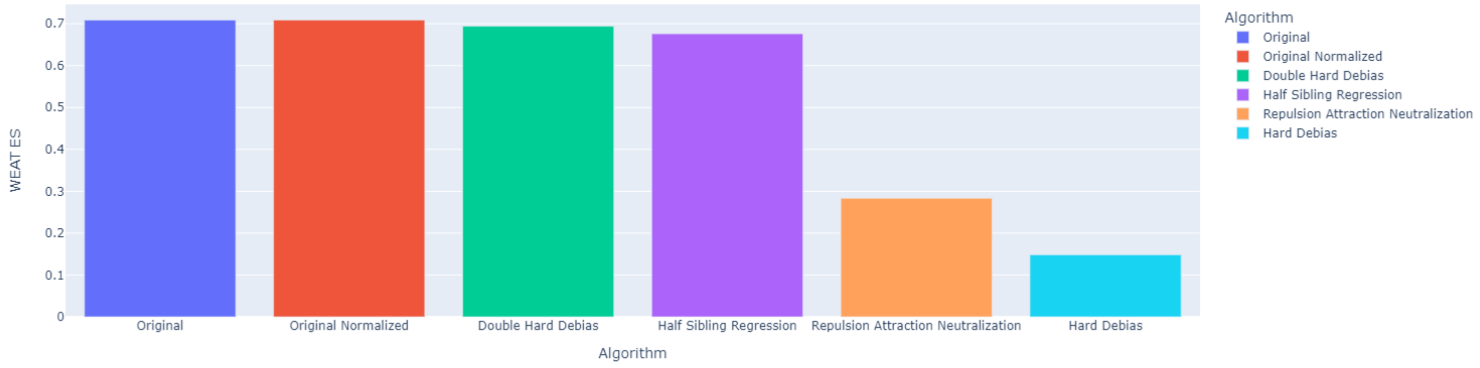

Figure A.13: WEAT ES results for Fasttext Wikipedia

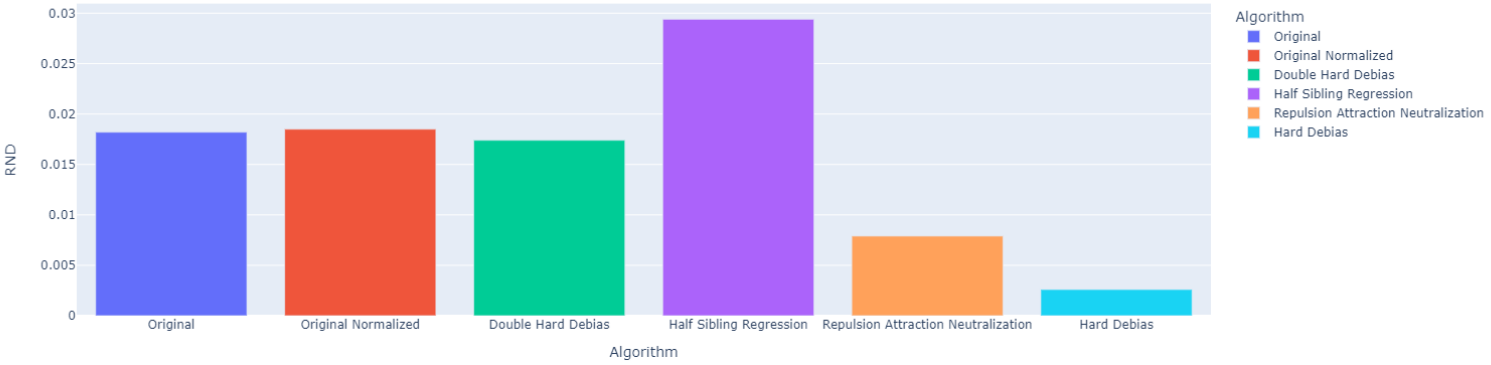

Figure A.14: RND results for Fasttext Wikipedia

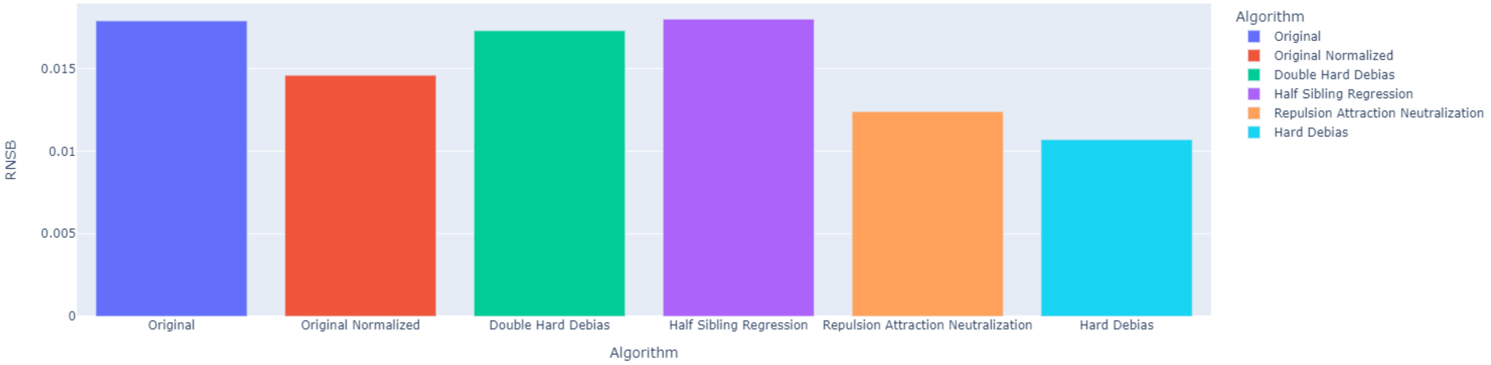

Figure A.15: RNSB results for Fasttext Wikipedia

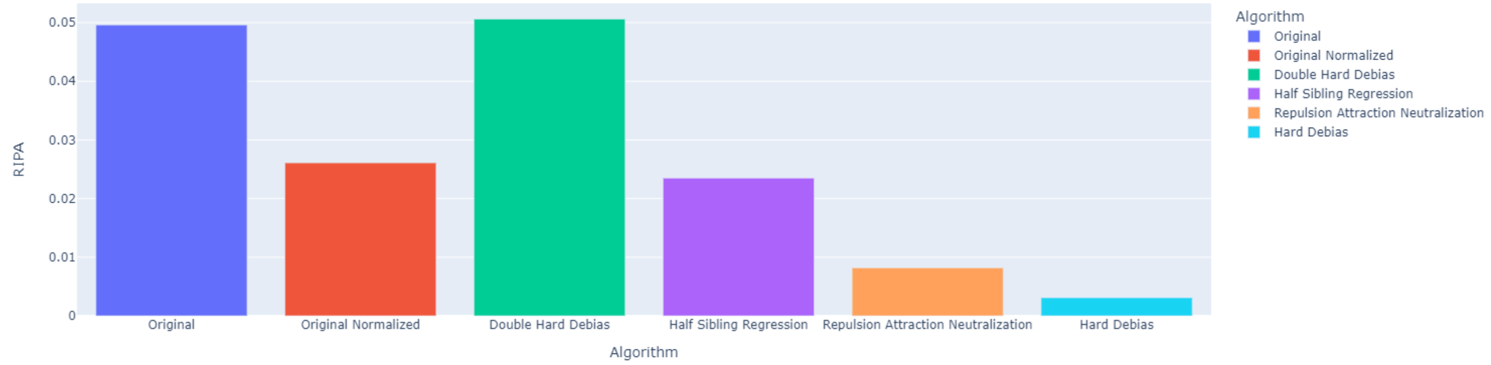

Figure A.16: RIPA results for Fasttext Wikipedia

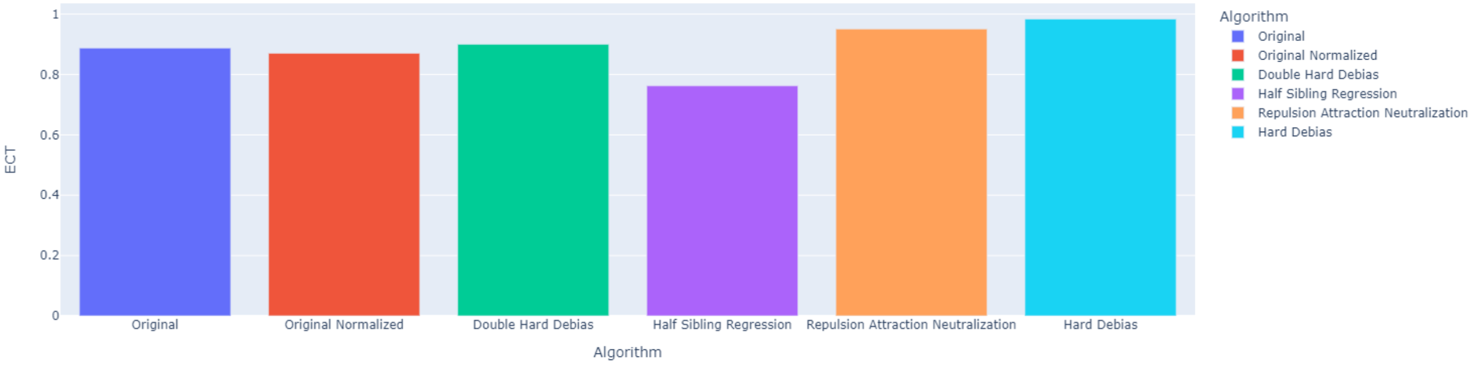

Figure A.17: ECT results for Fasttext Wikipedia

Glove-Wikipedia

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.845)	5 (0.656)	5 (0.183)	5 (0.075)	6 (0.225)	3 (0.823)
Original Normalized	6 (0.845)	6 (0.656)	3 (0.025)	2 (0.022)	4 (0.034)	4 (0.819)
Double Hard Debias	4 (0.573)	4 (0.569)	4 (0.134)	4 (0.064)	5 (0.158)	5 (0.814)
Half Sibling Regression	3 (0.229)	3 (0.557)	6 (0.191)	6 (0.207)	3 (0.032)	6 (0.55)
Repulsion Attraction Neutralization	2 (0.153)	2 (0.196)	2 (0.007)	3 (0.022)	2 (0.014)	2 (0.939)
Hard Debias	1 (0.089)	1 (0.136)	1 (0.006)	1 (0.019)	1 (0.005)	1 (0.968)

Table A.3: Bias metrics results and rankings for Glove-Wikipedia

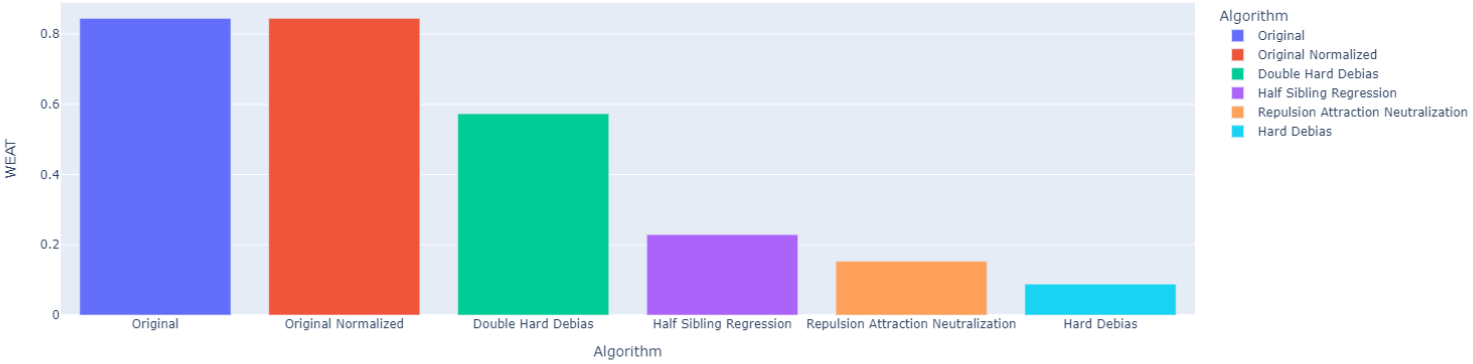

Figure A.18: WEAT results for Glove-Wikipedia

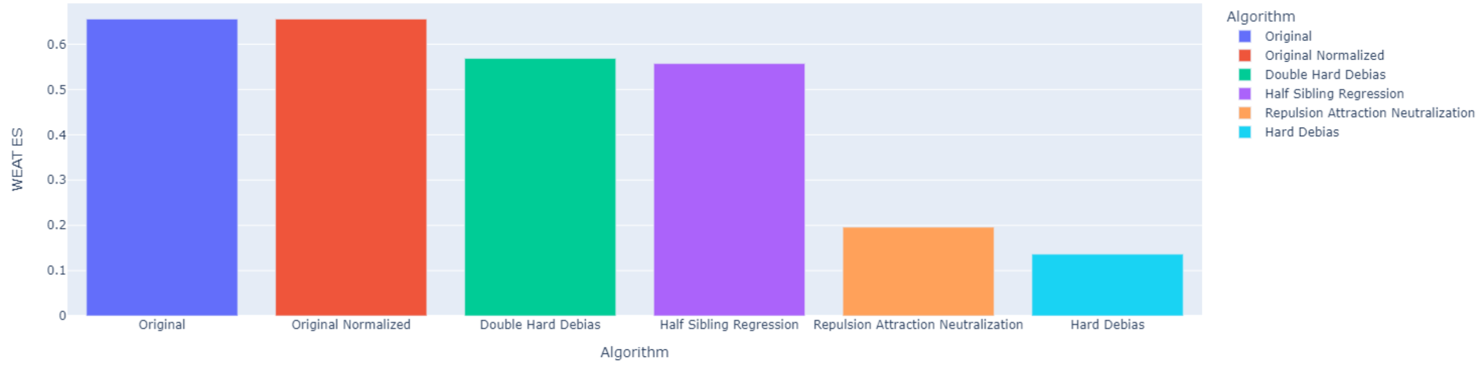

Figure A.19: WEAT ES results for Glove-Wikipedia

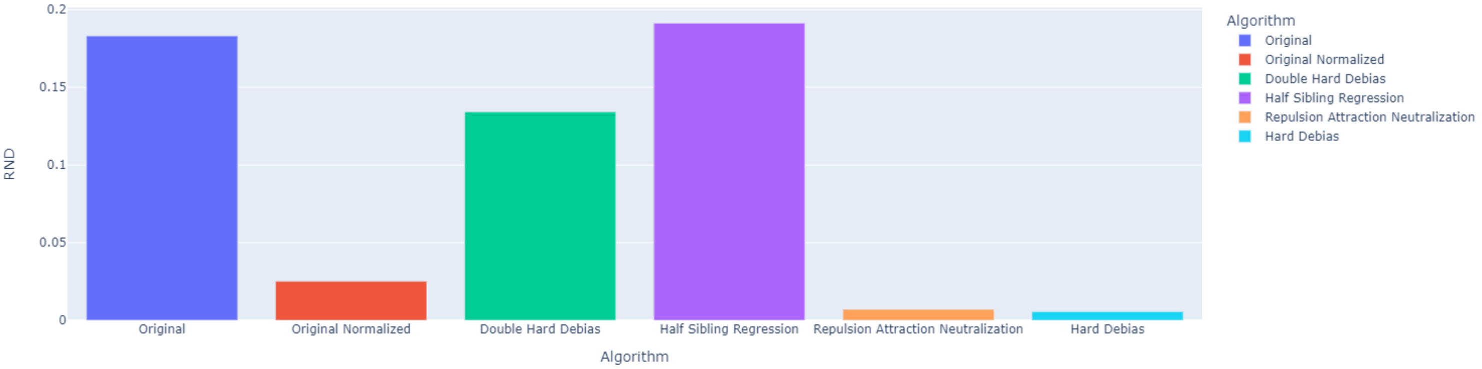

Figure A.20: RND results for Glove-Wikipedia

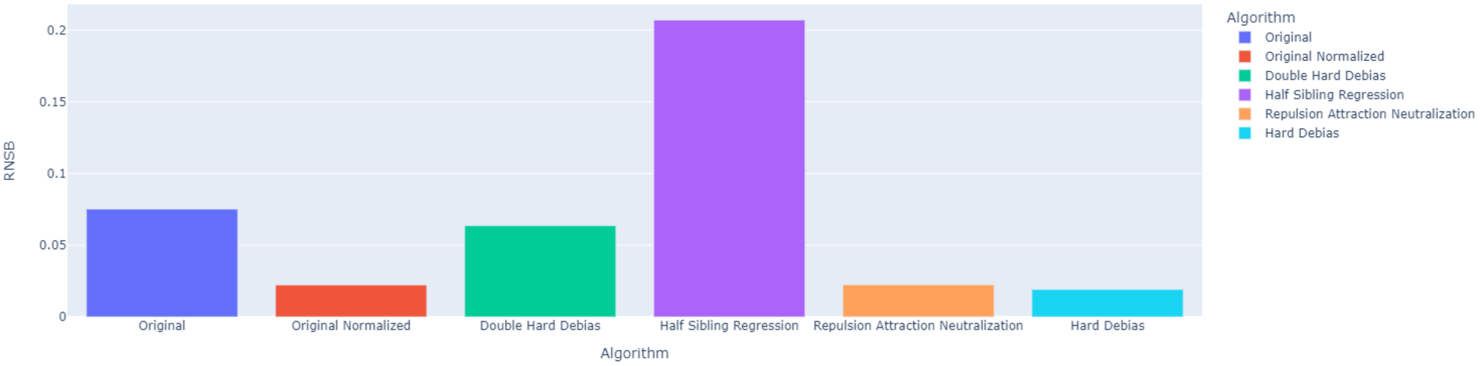

Figure A.21: RNSB results for Glove-Wikipedia

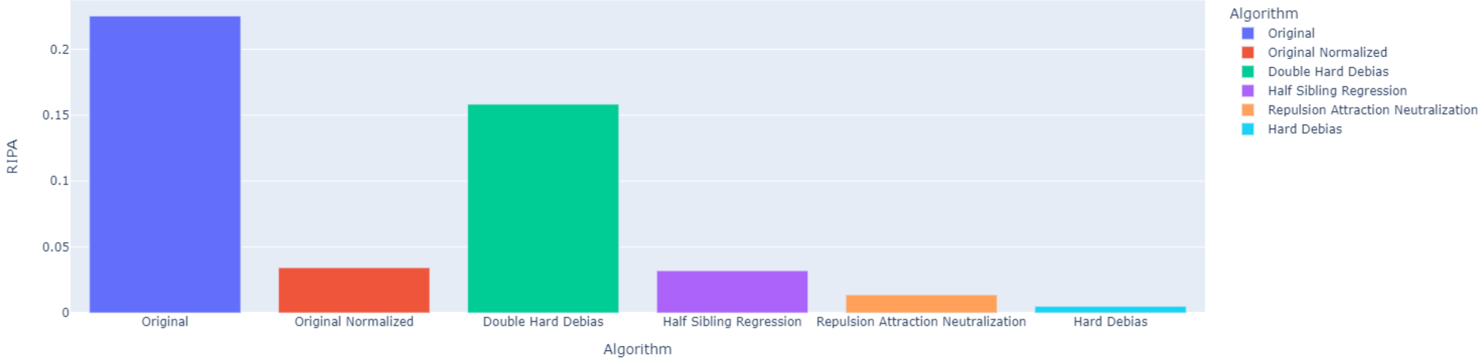

Figure A.22: RIPA results for Glove-Wikipedia

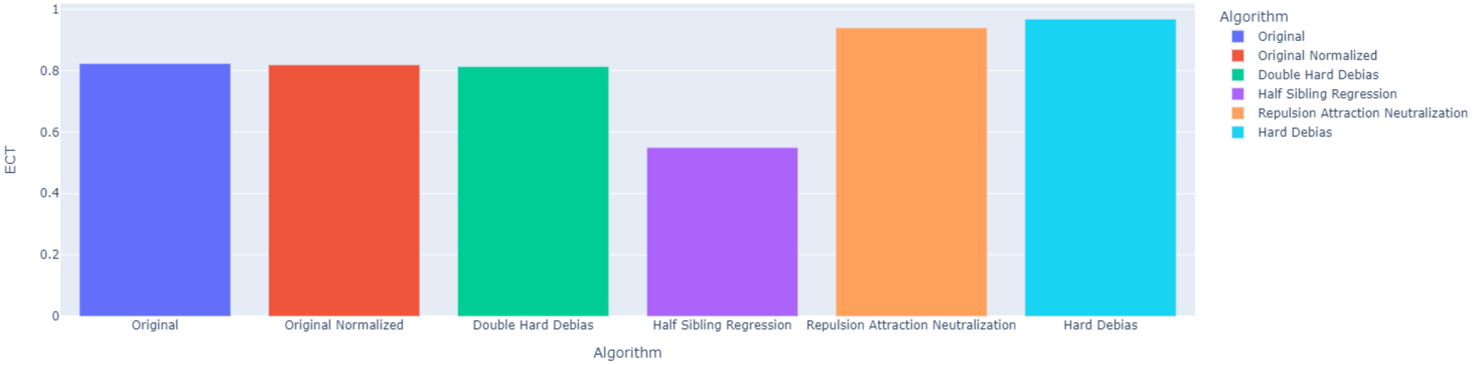

Figure A.23: ECT results for Glove-Wikipedia

Lexveccommoncrawl

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.533)	5 (0.762)	5 (0.042)	5 (0.022)	6 (0.059)	4 (0.796)
Original Normalized	6 (0.533)	6 (0.762)	3 (0.02)	3 (0.01)	3 (0.022)	5 (0.793)
Double Hard Debias	4 (0.432)	4 (0.678)	4 (0.037)	4 (0.021)	5 (0.047)	3 (0.843)
Half Sibling Regression	3 (0.263)	3 (0.563)	6 (0.042)	6 (0.039)	4 (0.026)	6 (0.594)
Repulsion Attraction Neutralization	2 (0.104)	2 (0.246)	2 (0.014)	2 (0.01)	2 (0.01)	2 (0.928)
Hard Debias	1 (0.067)	1 (0.18)	1 (0.002)	1 (0.008)	1 (0.003)	1 (0.976)

Table A.4: Bias metrics results and rankings for Lexveccommoncrawl

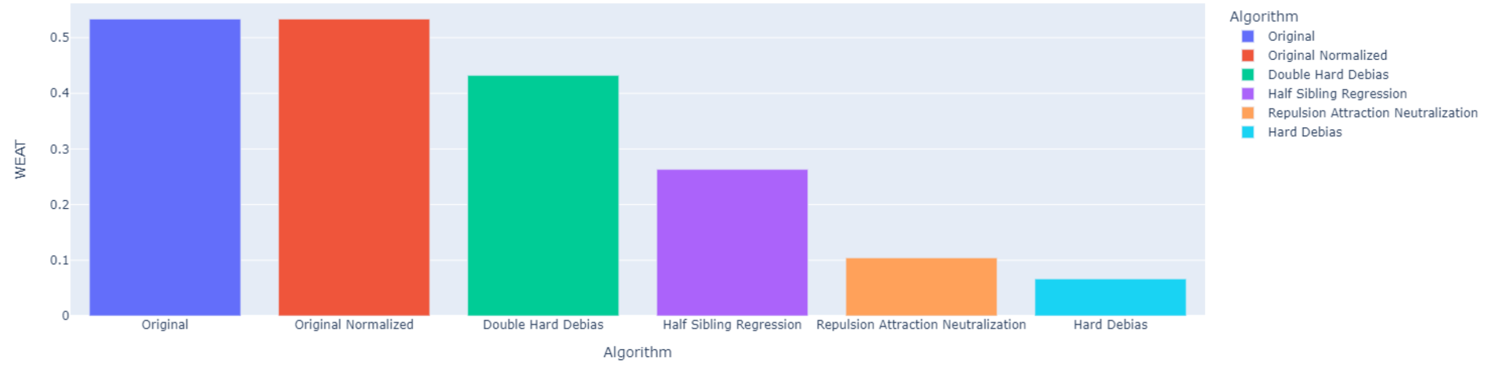

Figure A.24: WEAT results for Lexveccommoncrawl

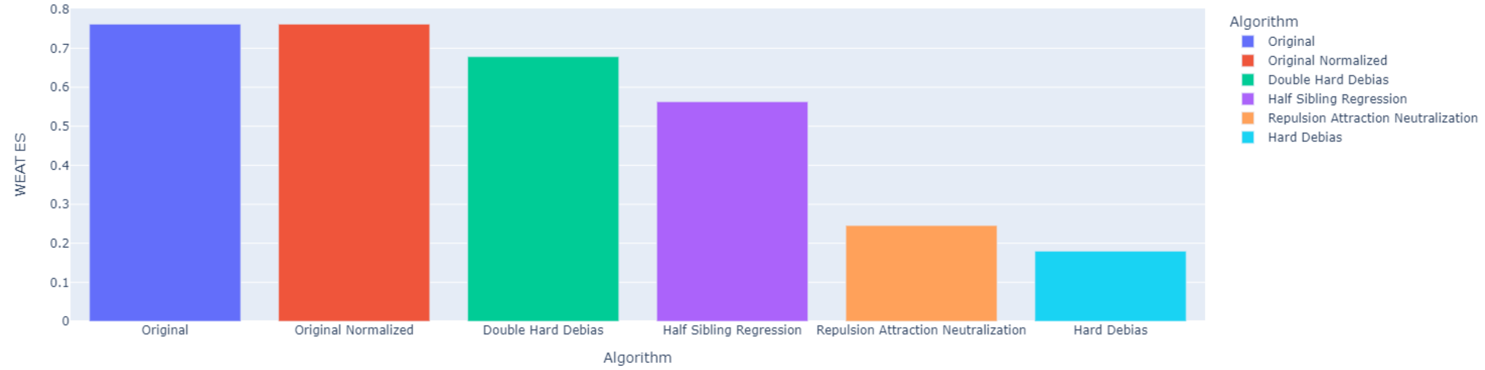

Figure A.25: WEAT ES results for Lexveccommoncrawl

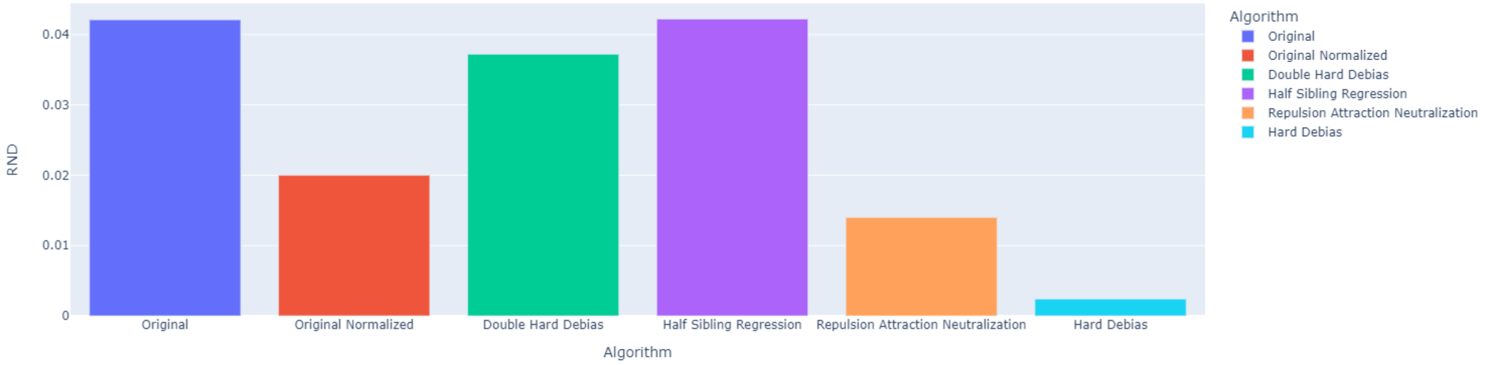

Figure A.26: RND results for Lexveccommoncrawl

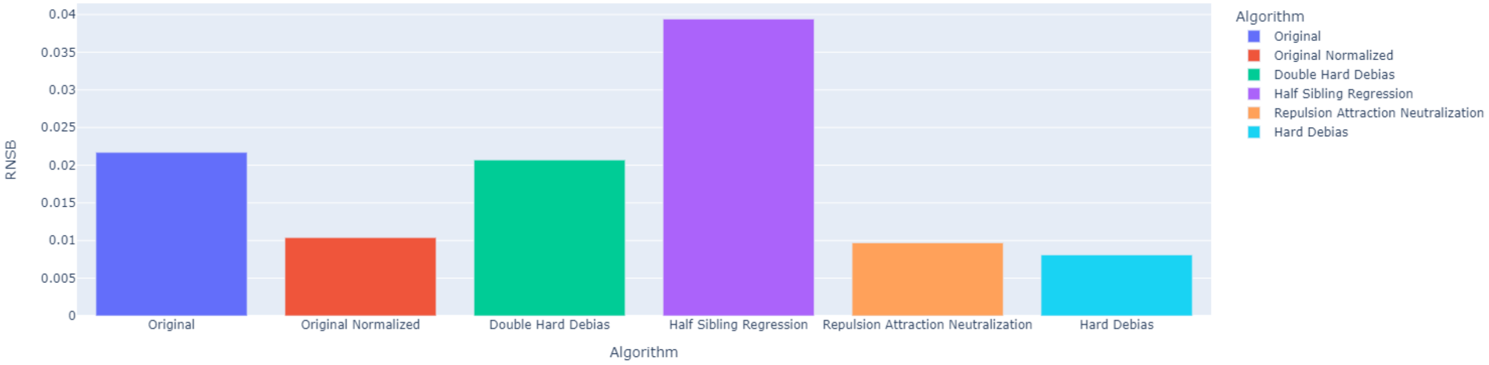

Figure A.27: RNSB results for Lexveccommoncrawl

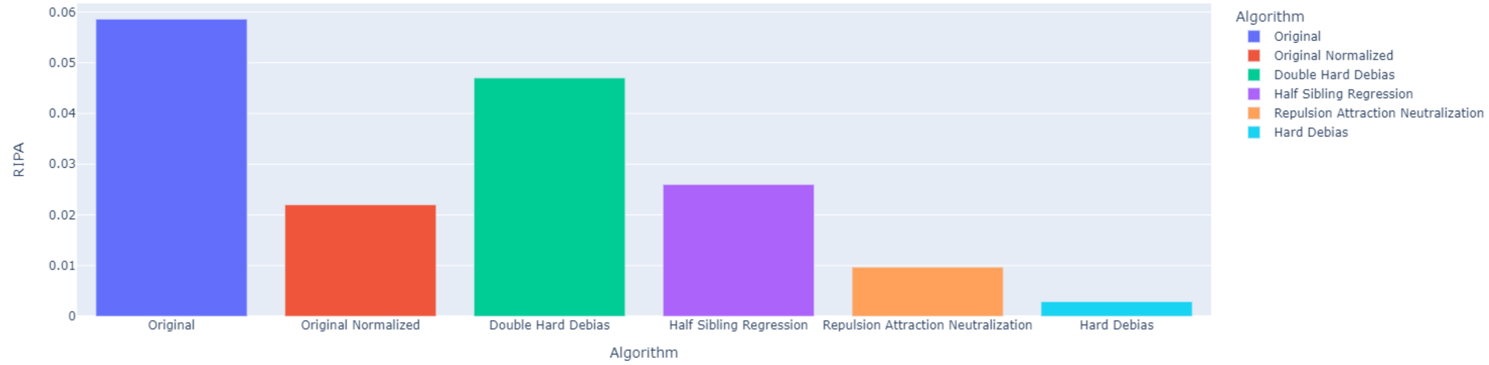

Figure A.28: RIPA results for Lexveccommoncrawl

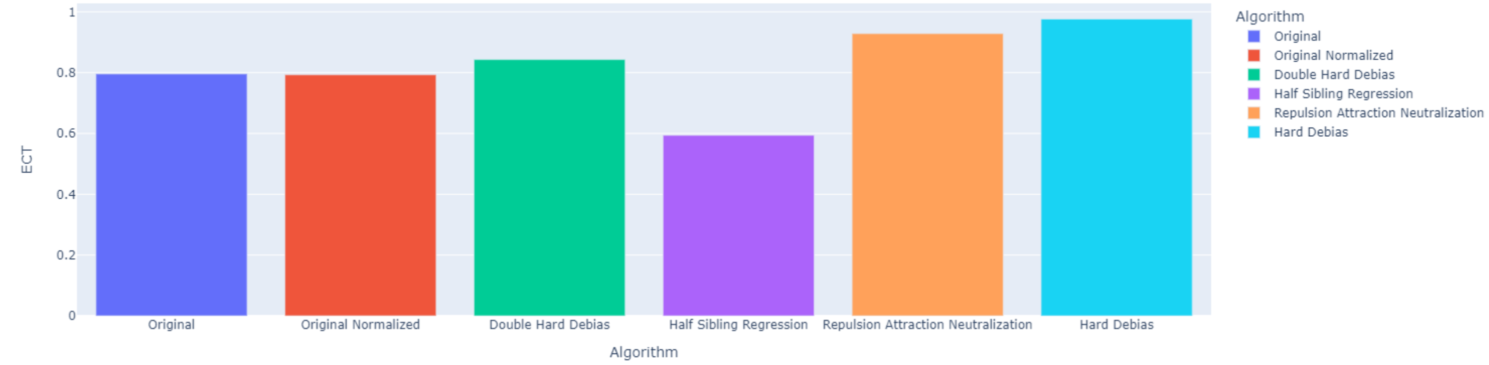

Figure A.29: ECT results for Lexveccommoncrawl

Word2vec Googlenews

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.83)	5 (0.941)	5 (0.084)	6 (0.034)	6 (0.108)	3 (0.768)
Original Normalized	6 (0.83)	6 (0.941)	3 (0.027)	3 (0.013)	4 (0.033)	5 (0.758)
Double Hard Debias	4 (0.782)	4 (0.93)	4 (0.083)	5 (0.033)	5 (0.101)	4 (0.765)
Half Sibling Regression	3 (0.22)	3 (0.488)	6 (0.114)	4 (0.03)	3 (0.027)	6 (0.459)
Repulsion Attraction Neutralization	2 (0.172)	2 (0.264)	2 (0.02)	2 (0.011)	2 (0.012)	2 (0.915)
Hard Debias	1 (0.086)	1 (0.18)	1 (0.003)	1 (0.01)	1 (0.004)	1 (0.985)

Table A.5: Bias metrics results and rankings for Word2vec Googlenews

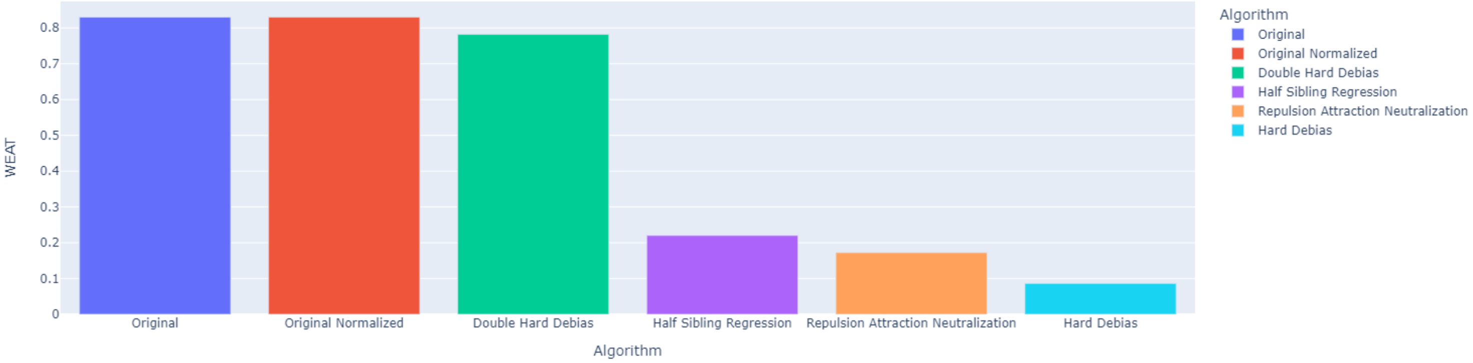

Figure A.30: WEAT results for Word2vec Googlenews

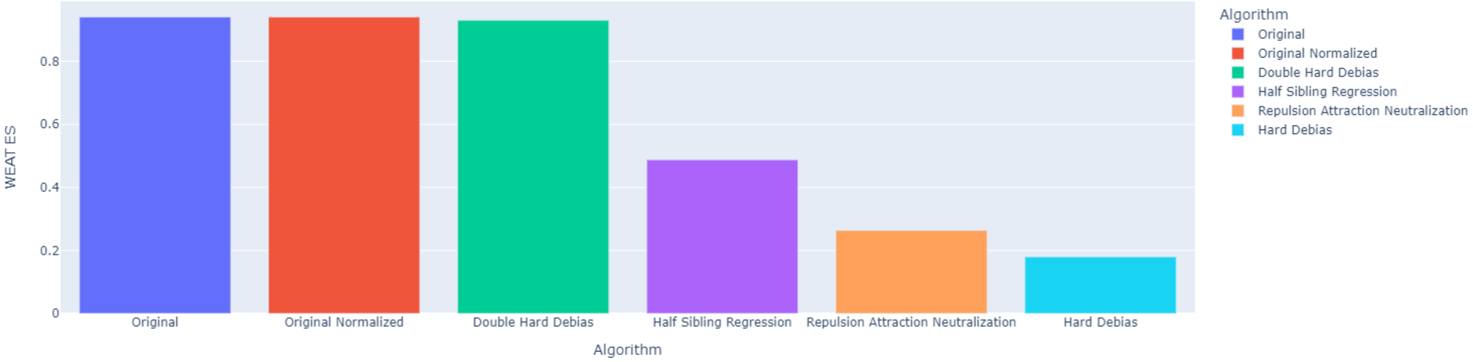

Figure A.31: WEAT ES results for Word2vec Googlenews

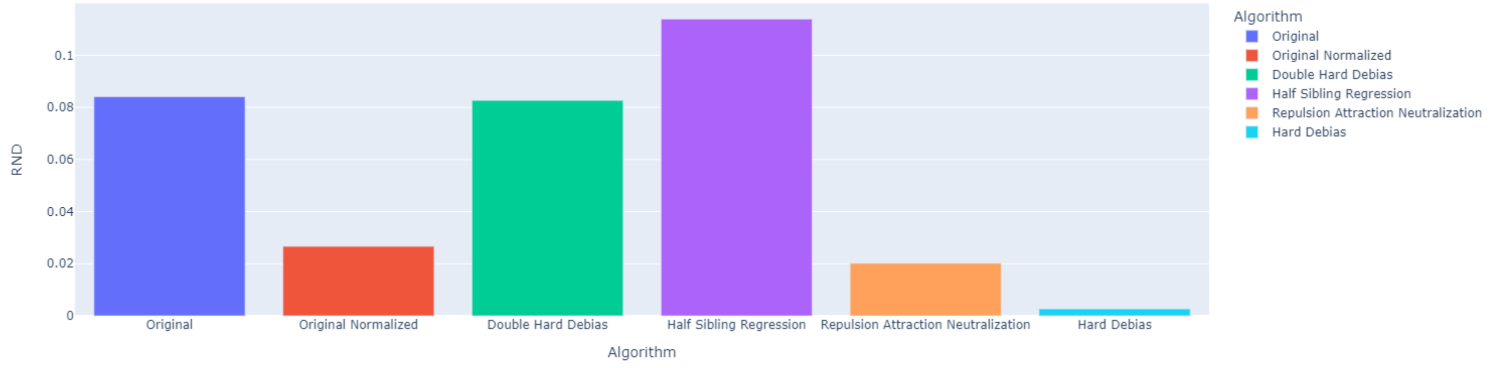

Figure A.32: RND results for Word2vec Googlenews

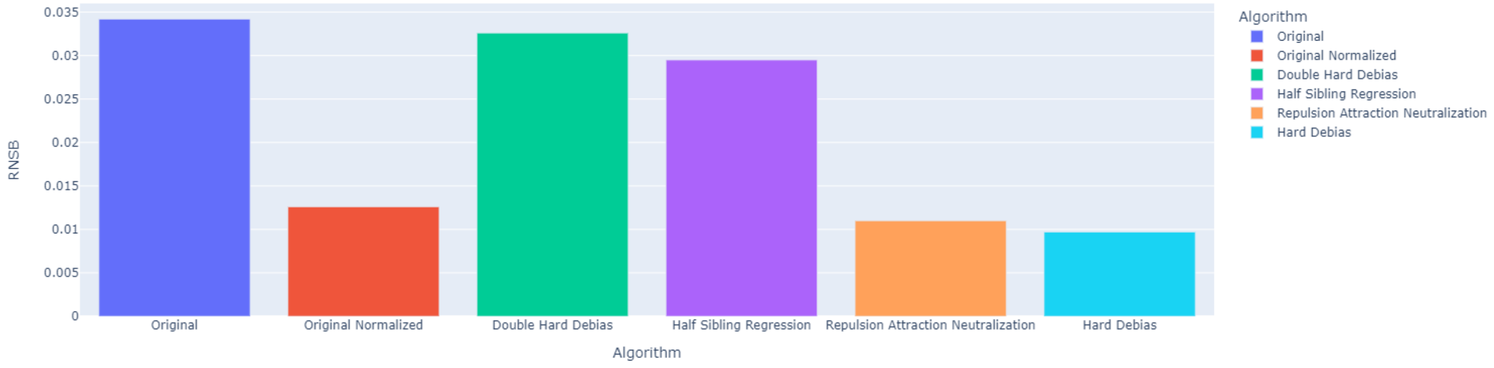

Figure A.33: RNSB results for Word2vec Googlenews

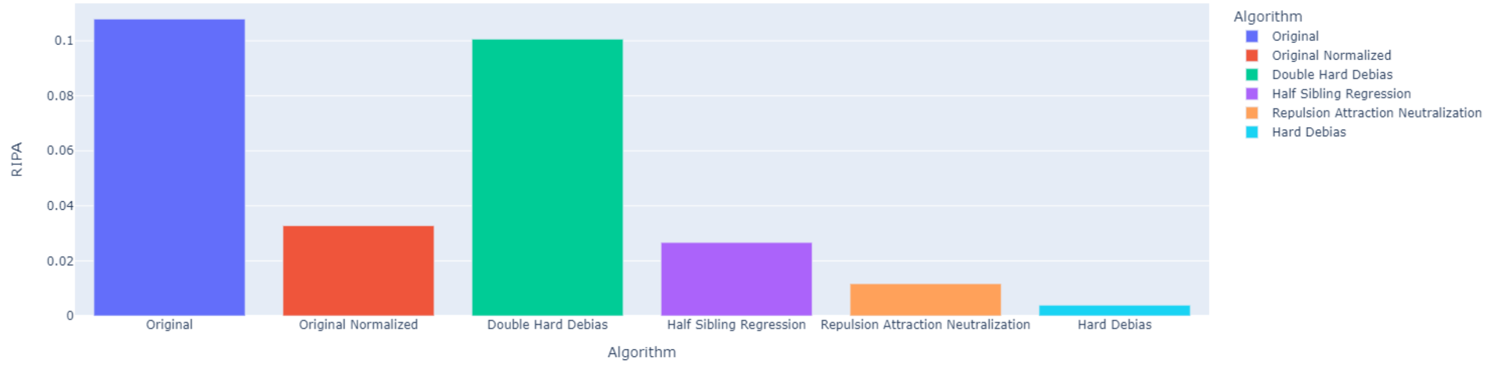

Figure A.34: RIPA results for Word2vec Googlenews

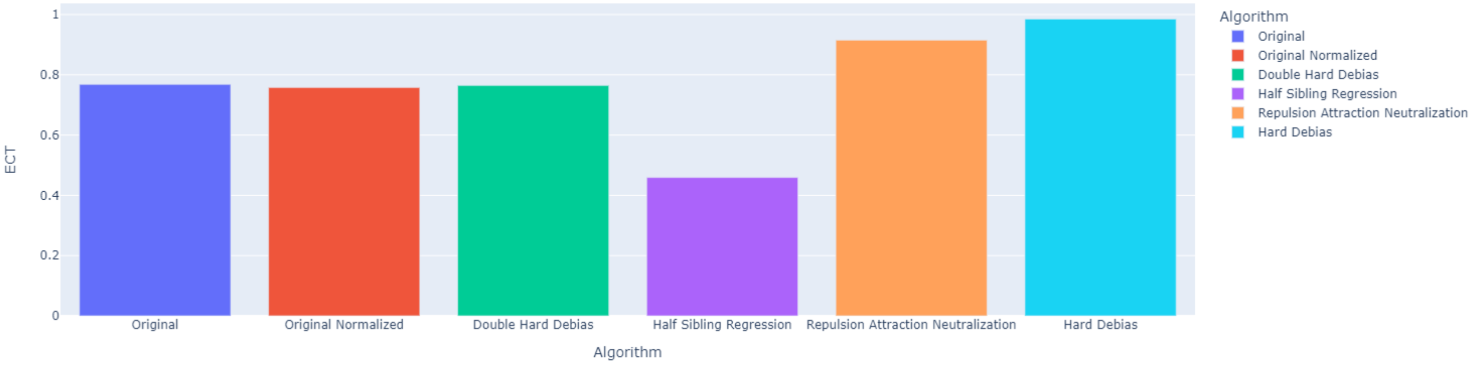

Figure A.35: ECT results for Word2vec Googlenews

A.2 WEB results

This section shows the results of evaluating the models using the WEB framework.

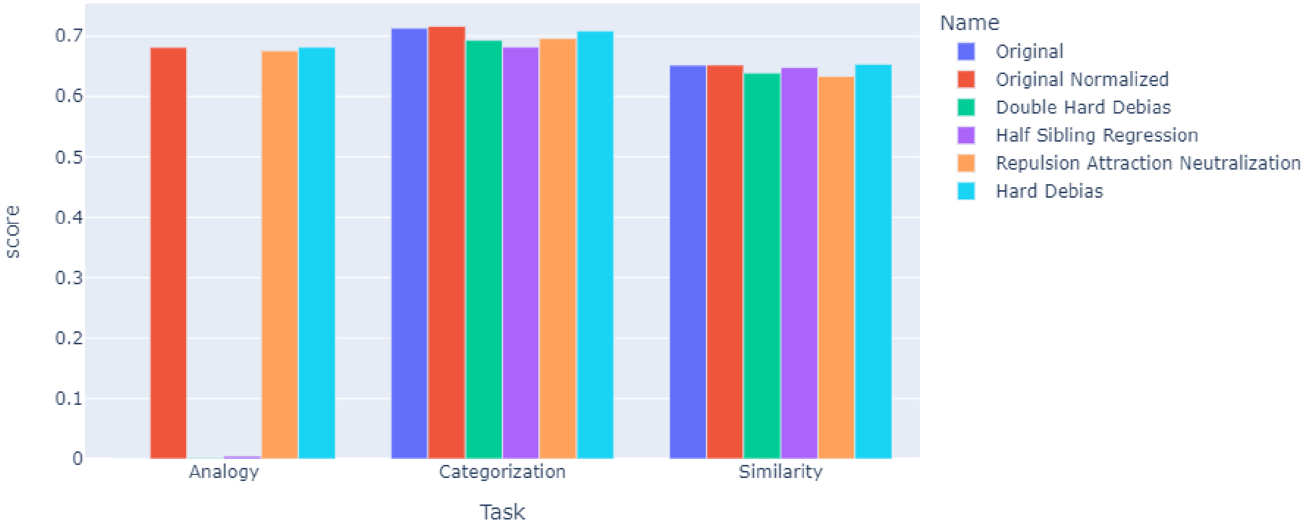

Figure A.36: Lexveccommoncrawl WEB results

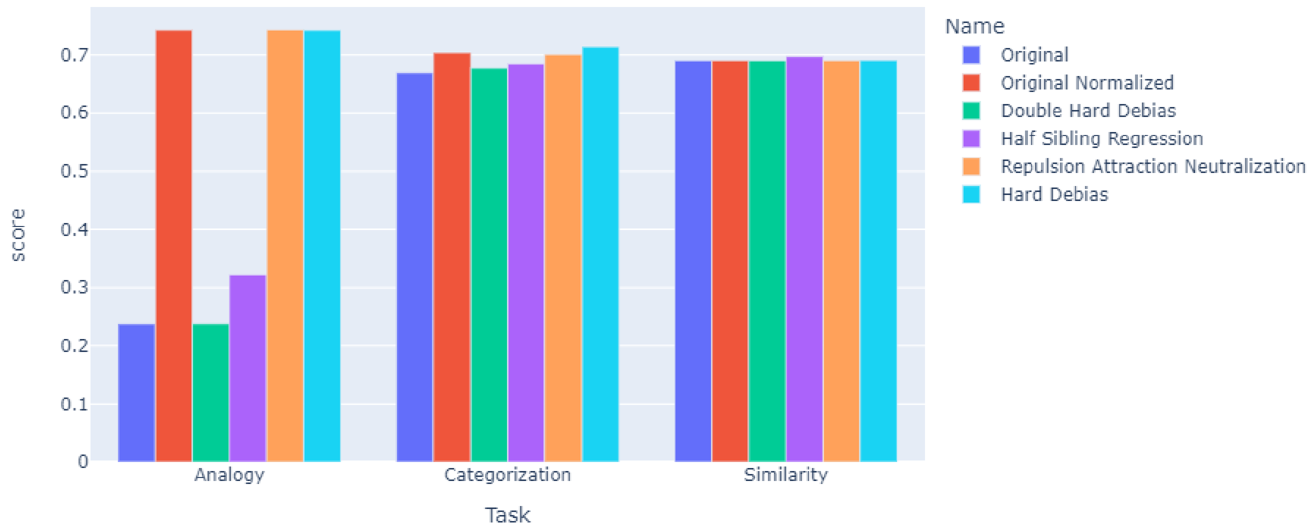

Figure A.37: Fasttext Wikipedia WEB results

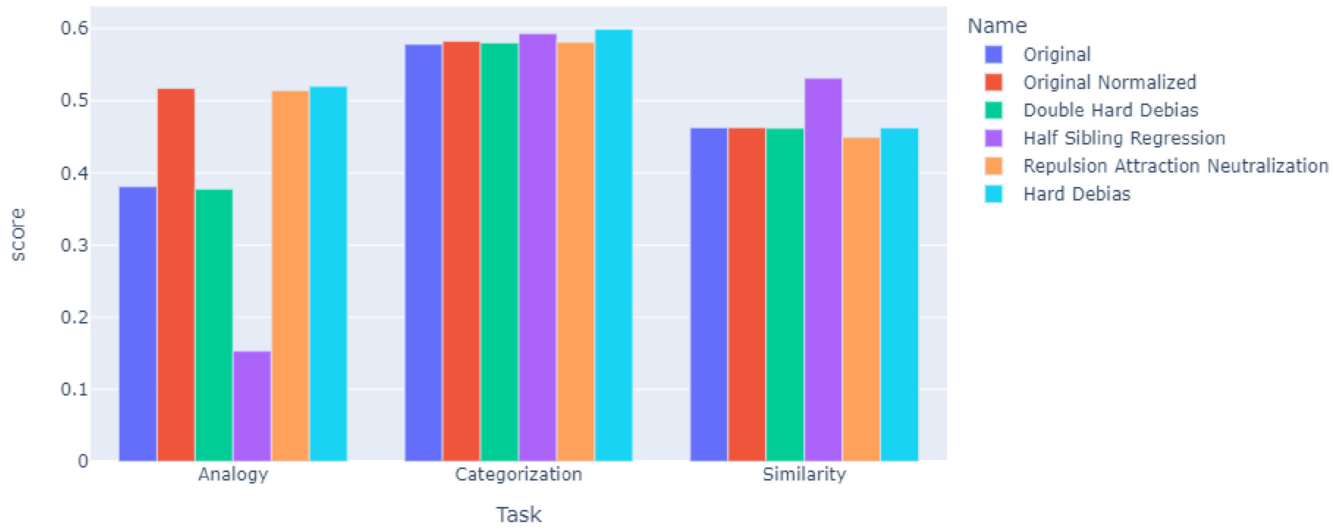

Figure A.38: Glove-Twitter WEB Results

A.3 Results including definitional pairs in DHD

This section includes the results tables and rankings when including the definitional pairs as part of the words to be debiased in DHD.

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.202)	5 (0.366)	5 (0.007)	4 (0.02)	5 (0.01)	5 (0.913)
Original Normalized	6 (0.202)	6 (0.366)	6 (0.007)	5 (0.02)	6 (0.01)	4 (0.913)
Double Hard Debias	3 (0.135)	3 (0.27)	2 (0.004)	3 (0.019)	3 (0.007)	2 (0.939)
Half Sibling Regression	4 (0.173)	4 (0.364)	4 (0.006)	2 (0.019)	4 (0.008)	6 (0.862)
Repulsion Attraction Neutralization	2 (0.121)	2 (0.228)	3 (0.004)	6 (0.021)	2 (0.006)	3 (0.915)
Hard Debias	1 (0.045)	1 (0.092)	1 (0.002)	1 (0.018)	1 (0.002)	1 (0.987)

Table A.6: Bias metrics results and rankings for Conceptnet including the definitional pairs in the debias process when executing DHD

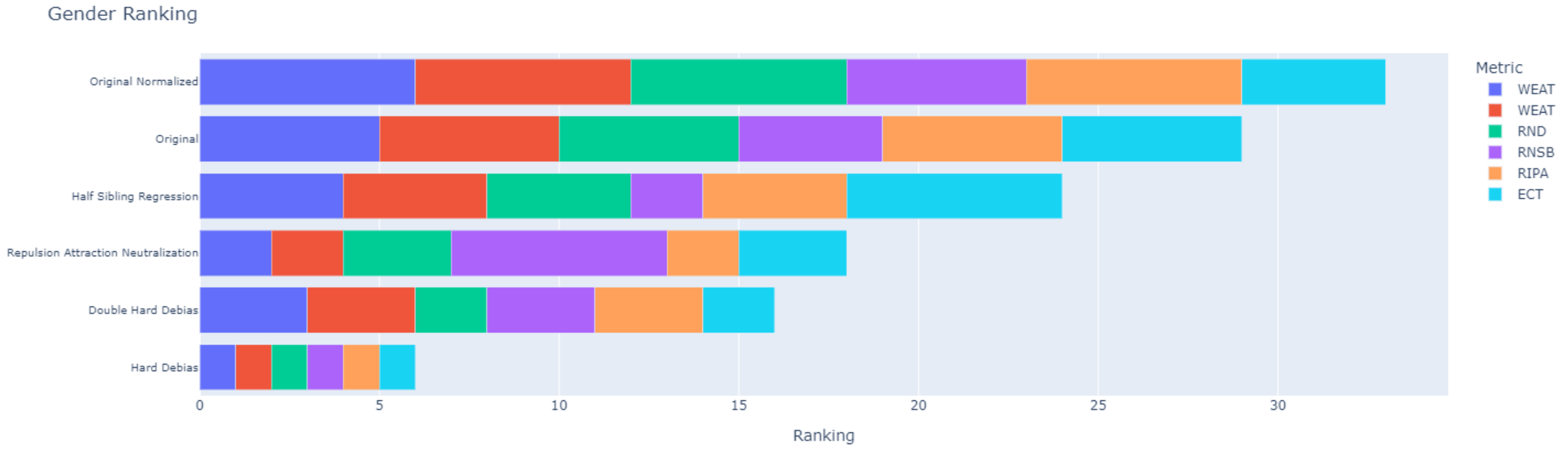

Figure A.39: Ranking showing the most biased models for Conceptnet including the definitional pairs in the debias process when executing DHD

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.468)	6 (0.709)	4 (0.018)	5 (0.018)	6 (0.05)	4 (0.888)
Original Normalized	6 (0.468)	5 (0.709)	5 (0.018)	3 (0.015)	5 (0.026)	5 (0.87)
Double Hard Debias	3 (0.3)	3 (0.503)	2 (0.007)	4 (0.015)	3 (0.021)	3 (0.944)
Half Sibling Regression	4 (0.436)	4 (0.676)	6 (0.029)	6 (0.018)	4 (0.023)	6 (0.762)
Repulsion Attraction Neutralization	2 (0.113)	2 (0.283)	3 (0.008)	2 (0.012)	2 (0.008)	2 (0.951)
Hard Debias	1 (0.055)	1 (0.148)	1 (0.003)	1 (0.011)	1 (0.003)	1 (0.984)

Table A.7: Bias metrics results and rankings for Fasttext Wikipedia including the definitional pairs in the debias process when executing DHD

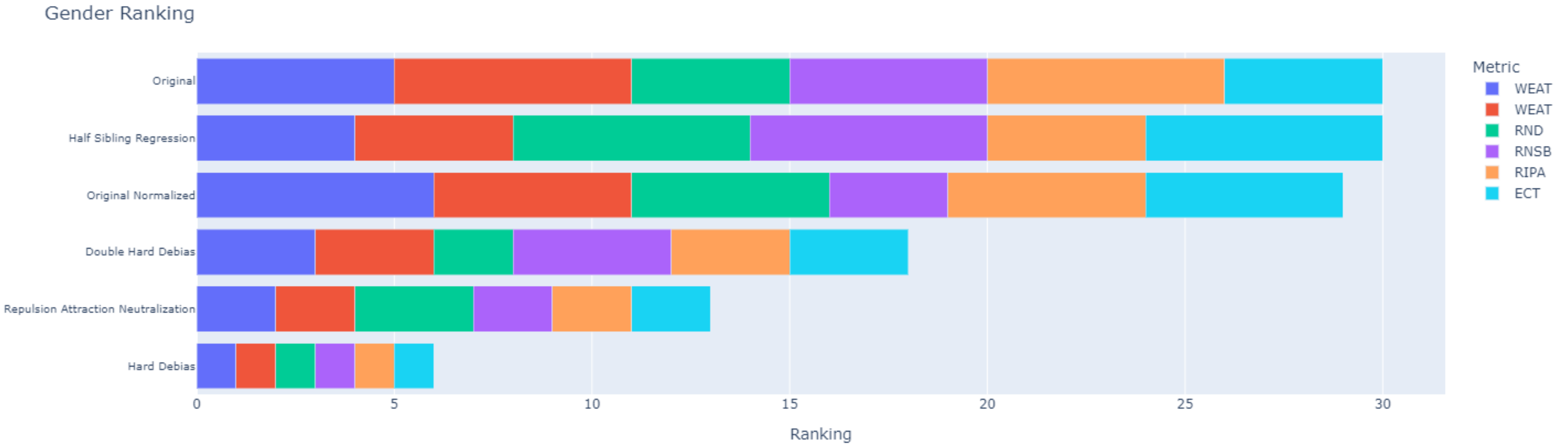

Figure A.40: Ranking showing the most biased models for Fasttext Wikipedia including the definitional pairs in the debias process when executing DHD

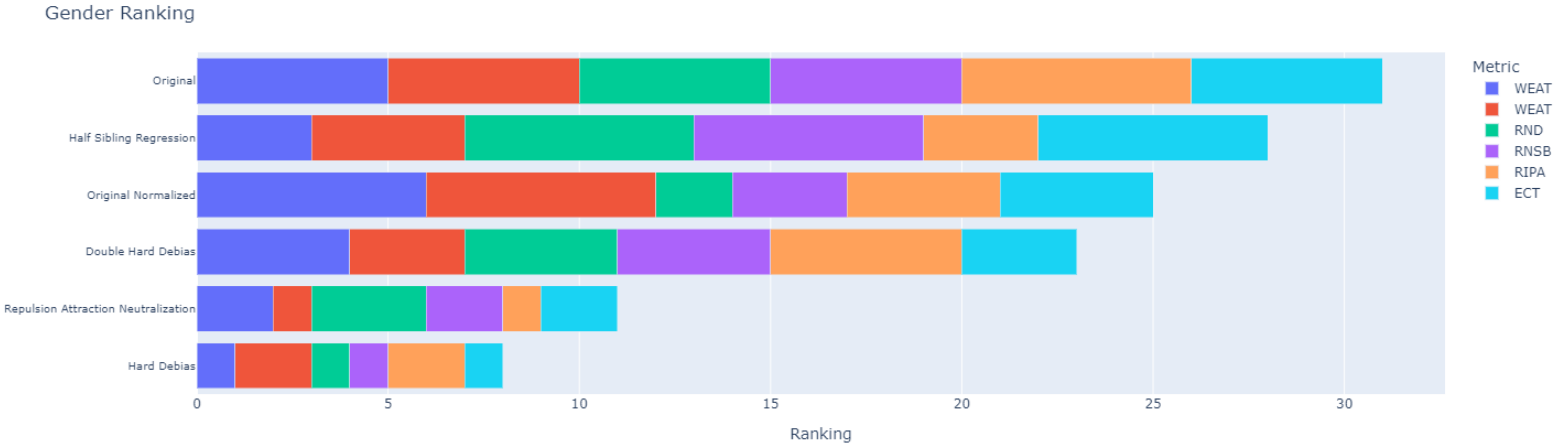

Figure A.41: Ranking showing the most biased models for Glove-Twitter including the definitional pairs in the debias process when executing DHD

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.845)	5 (0.656)	5 (0.183)	5 (0.074)	6 (0.225)	4 (0.823)
Original Normalized	6 (0.845)	6 (0.656)	3 (0.025)	3 (0.022)	4 (0.034)	5 (0.819)
Double Hard Debias	4 (0.442)	3 (0.401)	4 (0.08)	4 (0.071)	5 (0.156)	3 (0.93)
Half Sibling Regression	3 (0.229)	4 (0.557)	6 (0.191)	6 (0.208)	3 (0.032)	6 (0.55)
Repulsion Attraction Neutralization	2 (0.153)	2 (0.196)	2 (0.007)	2 (0.022)	2 (0.014)	2 (0.939)
Hard Debias	1 (0.089)	1 (0.136)	1 (0.006)	1 (0.019)	1 (0.005)	1 (0.968)

Table A.8: Bias metrics results and Rankings for Glove-Wikipedia including the definitional pairs in the debias process when executing DHD

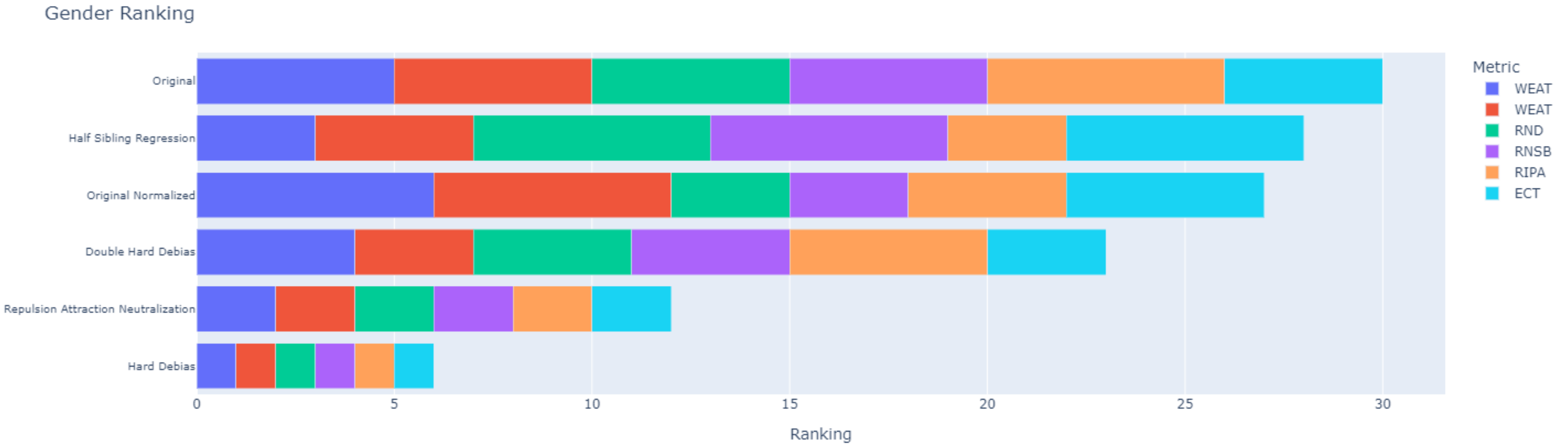

Figure A.42: Ranking showing the most biased models for Glove-Wikipedia including the definitional pairs in the debias process when executing DHD

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.533)	5 (0.762)	5 (0.042)	4 (0.021)	6 (0.059)	4 (0.796)
Original Normalized	6 (0.533)	6 (0.762)	3 (0.02)	3 (0.011)	3 (0.022)	5 (0.793)
Double Hard Debias	4 (0.289)	3 (0.42)	4 (0.026)	5 (0.027)	5 (0.037)	3 (0.866)
Half Sibling Regression	3 (0.263)	4 (0.563)	6 (0.042)	6 (0.039)	4 (0.026)	6 (0.594)
Repulsion Attraction Neutralization	2 (0.104)	2 (0.246)	2 (0.014)	2 (0.01)	2 (0.01)	2 (0.928)
Hard Debias	1 (0.067)	1 (0.18)	1 (0.002)	1 (0.008)	1 (0.003)	1 (0.976)

Table A.9: Bias metrics results and rankings for Lexveccommoncrawl including the definitional pairs in the debias process when executing DHD

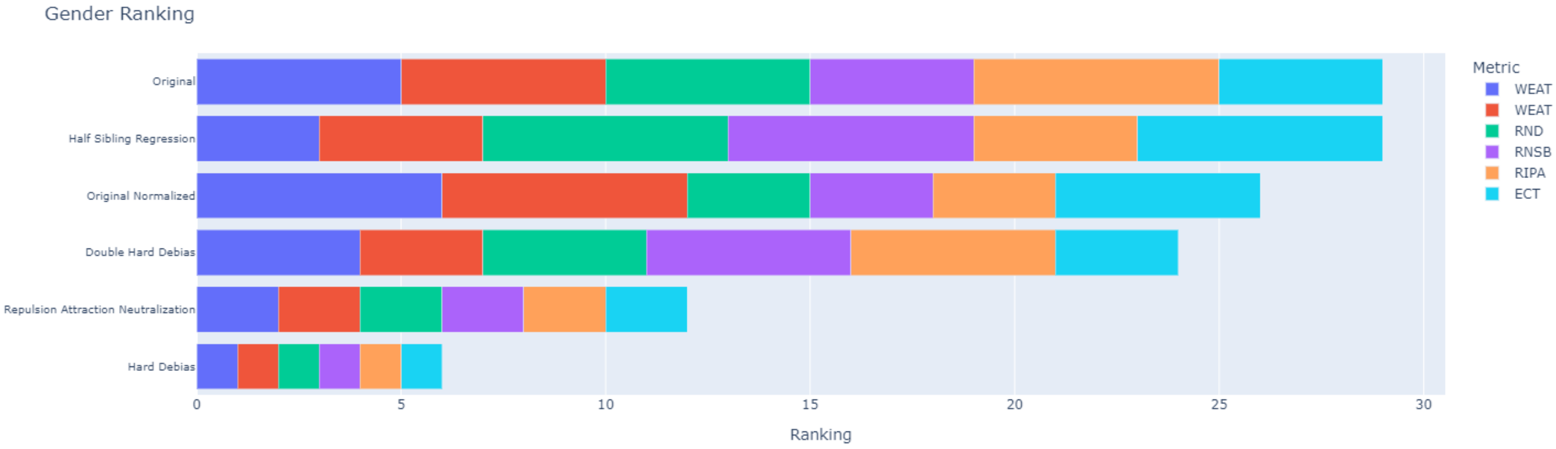

Figure A.43: Ranking showing the most biased models for Lexveccommoncrawl including the definitional pairs in the debias process when executing DHD

Model	WEAT	WEAT ES	RND	RNSB	RIPA	ECT
Original	5 (0.83)	5 (0.941)	5 (0.084)	6 (0.034)	6 (0.108)	4 (0.768)
Original Normalized	6 (0.83)	6 (0.941)	3 (0.027)	3 (0.013)	4 (0.033)	5 (0.758)
Double Hard Debias	4 (0.426)	4 (0.557)	4 (0.037)	4 (0.03)	5 (0.063)	3 (0.88)
Half Sibling Regression	3 (0.22)	3 (0.488)	6 (0.114)	5 (0.03)	3 (0.027)	6 (0.459)
Repulsion Attraction Neutralization	2 (0.172)	2 (0.264)	2 (0.02)	2 (0.011)	2 (0.012)	2 (0.915)
Hard Debias	1 (0.086)	1 (0.18)	1 (0.003)	1 (0.01)	1 (0.004)	1 (0.985)

Table A.10: Bias metrics results and rankings for Word2Vec-Googlenews including the definitional pairs in the debias process when executing DHD

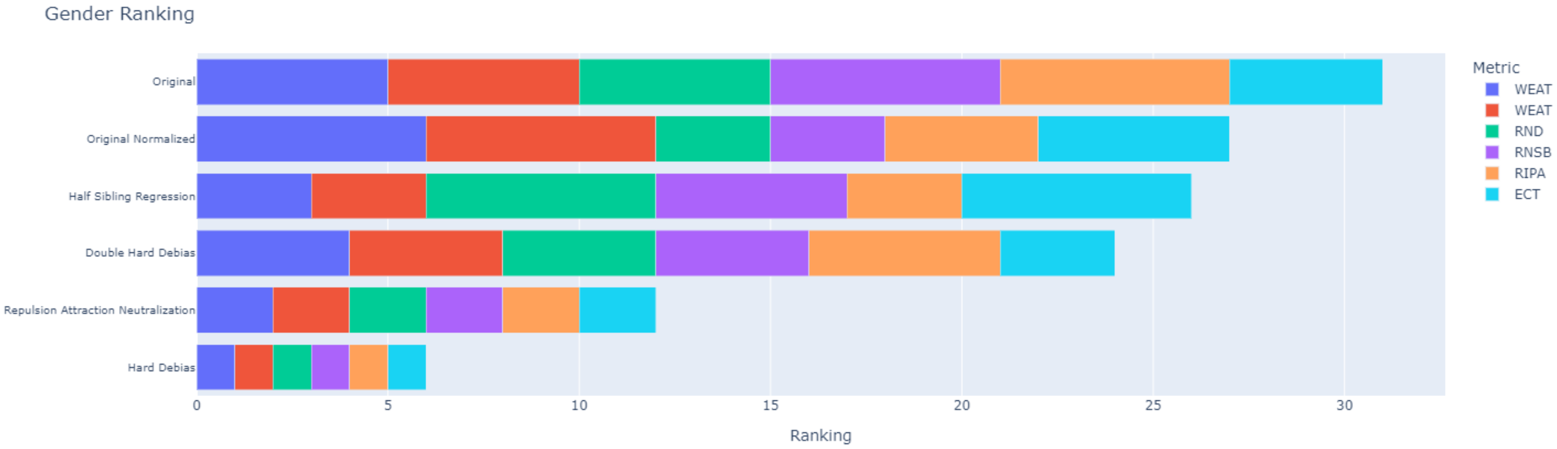

Figure A.44: Ranking showing the most biased models for Word2Vec-Googlenews including the definitional pairs in the debias process when executing DHD

Annexed B

Links and Wordsets

In this appendix we first present the links to the repositories where the algorithms are originally implemented. We then show the wordsets used as definitional pairs and gender specific (ignore parameters) when applying the algorithms to the models. Both of these sets are included in WEFE and are the ones used in Man is to Computer Programmer as Woman is to Homemaker? [7].

B.1 Links to the algorithms' original repositories

- *Hard debias*: `https://github.com/tolga-b/debiaswe`
- *Double Hard Debias*: `https://github.com/uvavision/Double-Hard-Debias`
- *Repulsion Attraction Neutralization*: `https://github.com/TimeTraveller-San/RAN-Debias`
- *Half Sibling Regression*: `https://github.com/KunkunYang/GenderBiasHSR`

B.2 Definitional pairs

[woman, man], [girl, boy], [she, he], [mother, father], [daughter, son], [gal, guy], [female, male], [her, his], [herself, himself], [Mary, John]

B.3 Gender Specific

[he, his, He, her, she, him, She, man, women, men, His, woman, spokesman, wife, himself, son, mother, father, chairman, daughter, husband, guy, girls, girl, Her, boy, King, boys, brother, Chairman, spokeswoman, female, sister, Women, Man, male, herself, Lions, Lady, brothers, dad, actress, mom, sons, girlfriend, Kings, Men, daughters, Prince, Queen, teenager, lady, Bulls, boyfriend, sisters, Colts, mothers, Sir, king, businessman, Boys, grandmother, grandfather, deer, cousin, Woman, ladies, Girls, Father, uncle, PA, Boy, Councilman, mum, Brothers, MA, males, Girl, Mom, Guy, Queens, congressman, Dad, Mother, grandson, twins, bull, queen, businessmen, wives, widow, nephew, bride, females, aunt, Congressman, prostate_cancer, lesbian, chairwoman, fathers, Son, moms, Ladies, maiden, granddaughter, younger_brother, Princess, Guys, lads, Ma, Sons, lion, Bachelor, gentleman, fraternity, bachelor, niece, Lion, Sister, bulls, husbands, prince, colt, salesman, Bull, Sisters, hers, dude, Spokesman, beard, filly, Actress, Him, princess, Brother, lesbians, councilman, actresses, Viagra, gentlemen, stepfather, Deer, monks, Beard, Uncle, ex_girlfriend, lad, sperm, Daddy, testosterone, MAN, Female, nephews, maid, daddy, mare, fiance, Wife, fiancee, kings, dads, waitress, Male, maternal, heroine, feminist, Mama, nieces, girlfriends, Councilwoman, sir, stud, Mothers, mistress, lions, estranged_wife, womb, Brotherhood, Statesman, grandma, maternity, estrogen, ex_boyfriend, widows, gelding, diva, teenage_girls, nuns, Daughter, czar, ovarian_cancer, HE, Monk, countrymen, Grandma, teenage_girl, penis, bloke, nun, Husband, brides, housewife, spokesmen, suitors, menopause, monastery, patriarch, Beau, motherhood, brethren, stepmother, Dude, prostate, Moms, hostess, twin_brother, Colt, schoolboy, eldest, brotherhood, Godfather, fillies, stepson, congresswoman, Chairwoman, Daughters, uncles, witch, Mommy, monk, viagra, paternity, suitor, chick, Pa, fiancé, sorority, macho, Spokeswoman, businesswoman, eldest_son, gal, statesman, schoolgirl, fathered, goddess, hubby, mares, stepdaughter, blokes, dudes, socialite, strongman, Witch, fiancée, uterus, grandsons, Bride, studs, mama, Aunt, godfather, hens, hen, mommy, Babe, estranged_husband, Fathers, elder_brother, boyhood, baritone, Diva, Lesbian, grandmothers, grandpa, boyfriends, feminism, countryman, stallion, heiress, queens, Grandpa, witches, aunts, semen, fella, granddaughters, chap, knight, widower, Maiden, salesmen, convent, KING, vagina, beau, babe, HIS, beards, handyman, twin_sister, maids, gals, housewives, Gentlemen, horsemen, Businessman, obstetrics, fatherhood, beauty_queen, councilwoman, princes, matriarch, colts, manly, ma, fraternities, Spokesmen, pa, fellas, Gentleman, councilmen, dowry, barbershop, Monks, WOMAN, fraternal, ballerina, manhood, Dads, heroines, granny, gynecologist, princesses, Goddess, yo, Granny, knights, eldest_daughter, HER, underage_girls, masculinity, Girlfriend, bro, Grandmother, grandfathers, crown_prince, Restless, paternal, Queen_Mother, Boyfriend, womens, Males, SHE, Countess, stepchildren, Belles, bachelors, matron, momma, Legs, maidens, goddesses, landlady, sisterhood, Grandfather, Fraternity, Majesty, Babes, lass, maternal_grandmother, blondes, ”maam”, Womens, divorcee, Momma, fathering, Effie, Lad, womanhood, missus, Sisterhood, granddad, Mens, papa, gf, sis, Husbands, Hen, womanizer, gynecological, stepsister, Handsome, Prince_Charming, BOY, stepdad, teen_ager, GIRL, dame, Sorority, beauty_pageants, raspy, harem, maternal_grandfather, Hes, deliveryman, septuagenarian, damsel, paternal_grandmother, paramour, paternal_grandparents, Nun, DAD, mothering, shes, ”HE_S”, Nuns, teenage_daughters, auntie, widowed_mother, Girlfriends, FATHER, virile, COUPLE, grandmas, Hubby, nan, vixen, Joan_Crawford, stepdaughters, endometrial_cancer, stepsons, loins, Grandson, Mitchells, erections, Matron, Fella, daddies, ter, Sweetie, Dudes, Princesses, Lads, lioness, Mamma, virility, bros, womenfolk, Heir, BROTHERS, manliness, patriarchs, earl, sisterly, Whore, Gynaecology, countess, convents, Oratory, witch_doctor, mamas, yah, aunty, aunties, Heiress, lasses, Breasts, fairer_sex, sorority_sisters, WIFE, Laurels, penile, nuh, mah, toms, mam, Granddad, premenopausal_women, Granddaddy, nana, coeds, dames, herdsman, Mammy, Fellas, Niece, menfolk, Grandad, bloods, Gramps, damsels, Granddaughter, mamma, concubine, Oros, Blarney, filial, broads, Ethel_Kennedy, ACTRESS, Tit, fianc, Hunk, Night_Shift, wifey, Lothario, Holy_Roman_Emperor, horse_breeder, grandnephew, Lewises, Muscular, feminist_movement, Sanan, womenâ_€_™, Fiancee, dowries, Carmelite, rah, n_roller, bay_filly, belles, Uncles, PRINCESS, womans, Homeboy, Blokes, Charmer, codger, Delta_Zeta, courtesans, grandaughter, SISTER, Highness, grandbabies, crone, Skip_Away, noblewoman, bf, jane, philandering_husband, Sisqo, mammy, daugher, director_Skip_Bertman, DAUGHTER, Royal_Highness, mannish, spinsters, Missus, madame, Godfathers, saleswomen, beaus, Risha, luh, sah, negligee, Womenâ_€_™, Hos, salesgirl, grandmom, Grandmas, Lawsons, countrywomen, Booby, darlin, Sheiks, boyz, wifes, Bayi, Il_Duce, â_€_œMy, fem, daugther, Potti, hussy, tch, Gelding, stemmed_roses, Damson, puh, Tylers, neice, Mutha, GRANDMOTHER, youse, spurned_lover, mae, Britt_Ekland, clotheshorse, Carlita_Kilpatrick, Cambest, Pretty_Polly, banshees, male_chauvinist, Arliss, mommas, maidservant, Gale_Harold, Little_Bo_Peep, Cleavers, hags, blowsy, Queen_Elizabeth_I., lassies, papas, BABE, ugly_ducklings, Jims, hellion, Beautician, coalminer, relaxin, El_Mahroug, Victoria_Secret_Angel, shepherdess, Mosco, Slacks, nanna, wifely, tomboys, LAH, hast, apo, Kaplans, milkmaid, Robin_Munis, John_Barleycorn, royal_highness, Meanie, NAH, trollop, roh, Jewess, Sheik_Hamad, mumsy, Big_Pussy, chil_dren, Aunt_Bea, basso, sista, girlies, nun_Sister, chica, Bubbas, massa, Southern_belles, Nephews, castrations, Mister_Ed, Grandsons, Calaf, Malachy_McCourt, Shamash, hey_hey, Harmen, sonofabitch, Donovans, Grannie, Kalinka, hisself, Devean, goatherd, hinds, El_Corredor, Kens, notorious_womanizer, goh, Mommas, washerwoman, Samaira, Coo_Coo, Governess, grandsire, PRINCE_WILLIAM, gramma, him.He, Coptic_priest, Corbie, Kennys, thathe, Pa_Pa, Bristols, Hotep, snowy_haired, El_Prado_Ire, Girl_hitmaker, Hurleys, St._Meinrad, sexually_perverted, authoress, Prudie, raven_haired_beauty, Bonos, domestic_shorthair, brothas, nymphet, Neelma, Seita, stud_muffin, St._Judes, yenta, bare_shouldered, Pinkney_Sr., PRINCE_CHARLES, Bisutti, sistas, Blanche_Devereaux, Momoa, Quiff, Scotswoman, balaclava_clad_men, Louis_Leakey, dearie, vacuum_cleaner_salesman, grandads, postulant, SARAH_JESSICA_PARKER, AUNT, Prince_Dauntless, Dalys, Darkie, Czar_Nicholas, Lion_Hearted, Boy_recliner, baby_mamas, giantess, Lawd, GRANNY, fianc_e, Bilqis, WCTU, famly, Ellas, feminazis, Pentheus, MAMAS, Town_Criers, Saggy, youngman, grandam, divorcé, bosomed, roon, Simmentals, eponymous_heroine, LEYLAND, ”REE”, ”caint”, Evelynn, ”WAH”, sistah, Horners, Elsie_Poncher, Coochie, rat_terriers, Limousins, Buchinski, Schicchi, Carpitcher, Khwezi, ”HAH”, Shazza, Mackeson, ”ROH”, kuya, novice_nun, Shei, Elmasri, ladykiller, 6yo, Yenta, SHEL, pater, Souse, Tahirah, comedian_Rodney_Dangerfield, Shottle, carryin, Sath, ”faafafine”, royal_consort, hus_band, maternal_uncles, dressing_provocatively, dreamgirl, millionaire_industrialist, Georgie_Girl, Must_Be_Obeyed, joh, Arabian_stallion, ahr, mso_para_margin_0in, ”SOO”, Biddles, Chincoteague_Volunteer_Fire, Lisa_Miceli, gorgeous_brunette, fiancŽ, Moved_fluently, Afternoon_Deelites, biker_dude, Vito_Spatafore, MICK_JAGGER, Adesida, Reineman, witz, Djamila, Glenroe, daddys, Romanzi, gentlewomen, Dandie_Dinmont_terrier, Excess_Ire, By_SYVJ_Staff, zan, CONFESSIONS, Magees, wimmin, tash, Theatrical_Ire, Prince_Charmings, chocolate_eclair, bron, daughers, Felly, fiftyish, Spritely, GRANDPA, distaffer, Norbertines, ”DAH”, leader _Muammar_Gadaffi, swains, Prince_Tomohito, Honneur, Soeur, jouster, Pharaoh_Amenhotep_III, QUEEN_ELIZABETH_II, ”Neer”, Galileo_Ire, Fools_Crow, Lannisters, Devines, gonzales, columnist_Ann_Landers, Moseleys, hiz, busch, roastee, toyboys, Sheffields, grandaunt, Galvins, Giongo, geh, flame_haired_actress, Grammarian, Greg_Evigan, frontierswoman, Debele, rabs, nymphets, aai, BREE, Shaqs, ZAY, pappa, Housa, refrigerator_repairman, artificial_inseminations, chickie, Rippa, teenager_Tracy_Turnblad, homebred_colt, Abigaille, hen_pecked_husband, businesman, her.She, Kaikeyi, Stittsworth, self_proclaimed_redneck, Khella, NeW, Evers_Swindell, Asmerom_Gebreselassie, Boy_recliners, Cliff_Claven, Legge_Bourke, Costos, ”d_honneur”, sistahs, Cabble, sahn, CROW_AGENCY_Mont, jezebel, Harrolds, ROSARIO_DAWSON, INXS_frontman_Michael_Hutchence, Gursikh, Dadas, VIAGA, keen_horsewoman, Theodoric, Eldery, lihn, Alice_Kramden, Santarina, radical_cleric_al_Sadr, Curleys, ”SY”, Fidaa, Saptapadi, Actor_Sean_Astin, Kellita_Smith, Doly, Libertina, Money_McBags, Chief_Bearhart, choirgirl, chestnut_stallion, VIGRA, BY_JIM_McCONNELL, Sal_Vitale, Trivia_buffs, kumaris, fraternal_lodge, galpals, Borino_Quinn, lina, LATEST_Rapper, Bezar, Manro, bakla, Grisetti, blond_bimbo, spinster_aunt, gurls, hiswife, paleface, Charlye, hippie_chicks, Khalifas, Picture_JUSTIN_SANSON, Hepburns, yez, ALDER, Sanussi, Lil_Sis, McLoughlins, Barbra_Jean, Lulua, thatshe, actress_Shohreh_Aghdashloo, SIR_ANTHONY_HOPKINS, Gloddy, ”ZAH”, ”ORANGE_S”, Danielle_Bimber, grandmum, Kulkis, Brazington, Marisa_Lenhard_CFA, SIR_JOHN, Clareman, Aqila, Heavily_tattooed, Libbys, thim, elocutionist, submissives, Inja, rahm, Agnes_Gooch, fake_tits, nancy_boys, Swaidan, ”SHAH”, ”ainta_bed”, Shumail_Raj, Duchesse, diethylstilbestrol_DES, colt_foal, unfaithful_lover, Maseri, nevah, SAHN, Barths, Toughkenamon, GUEST_STARS, him.But, Donna_Claspell, gingham_dresses, Massage_Parlour, wae, Wasacz, Magistra, vihl, Smriti_Iraani, boyish_haircut, workingwoman, borthers, Capuchin_friars, Nejma, yes_sirs, bivocational_pastor, Grafters, HOPWOOD, Nicknamed_Godzilla, yos, Berkenfield, Missis, sitcom_Designing_Women, Kafoa, trainer_Emma_Lavelle, sadomasochistic_dungeon, iht, desperates, predessor, wolf_cub, indigenous_Peruvians, Livia_Soprano, troh, colt_sired, BOND_HILL, ihl, Drydens, rahs, Piserchia, Sonny_Corinthos, bankrobber, Fwank, feisty_redhead, booze_guzzling, COOPERS, ”actress_Qorianka_Kilcher”, Cortezar, twe, Jacoub, Cindy_Iannarelli, Hell_Raiser, Fondly_referred, Bridal_Shoppe, Noleta, Christinas, IAGRA, LaTanya_Richardson, Sang_Bender, Assasins, sorrel_gelding, septugenarian, Hissy, Muqtada_al_Sadr_mook, Pfeni, MADRID_AFX_Banco_Santander, tuchis, LeVaughn, Gadzicki, transvestite_hooker, Fame_jockey_Laffit, nun_Sister_Mary, SAMSONOV, Mayflower_Madam, Shaque, well.He, Trainer_Julio_Canani, sorrel_mare, minivehicle_joint_venture, wife_Dwina, ”Aasiya_AH_see”, Baratheon, ”Rick_OShay”, Mammies, goatie, Nell_Gwynne, charmingly_awkward, Slamma, DEHL, Lorenzo_Borghese, ALMA_Wis., Anne_Scurria, father_Peruvians_alternately, JULIE_ANDREWS, Slim_Pickins, Victoria_Secret_stunner, ”BY”, Sanam_Devdas, pronounced_luh, Pasha_Selim, , rson, maternal_grandmothers, IOWA_CITY_Ia, Madame_de_Tourvel, ”JAY”, Sheika_Mozah_bint_Nasser, Hotsy_Totsy, ”D_Ginto”, singer_Johnny_Paycheck, uterine_ prolapse_surgery, SCOTTDALE_Pa., AdelaideNow_reports, Marcus_Schenkenberg, Clyse, Obiter_Dicta, comic_Sam_Kinison, bitties, ROCKVILLE_Ind., swimsuit_calendars, Decicio_Smith, Ma_ma, Rie_Miyazawa, celibate_chastity, gwah, ”ZAY”, HER_Majesty, Defrere, Las_Madrinas, __, Bea_Hamill, ARCADIA_Calif._Trainer, Bold_Badgett, stakes_victress, Hoppin_Frog, Narumiya, Flayfil, hardman_Vinnie_Jones, Marilyn_Monroe_lookalike, Kivanc_Tatlitug, Persis_Khambatta, SINKING_SPRING_Pa., len_3rd, DEAR_TRYING, Farndon_Cheshire, Krishna_Madiga, daughter_Princess_Chulabhorn, Marshall_Rooster_Cogburn, Kitty_Kiernan, Yokich, Jarou, Serdaris, ee_ay, Montifiore, Chuderewicz, Samuel_Le_Bihan, filly_Proud_Spell, Umm_Hiba, pronounced_koo, Sandy_Fonzo, ”KOR”, Fielder_Civil_kisses, Federalsburg_Maryland, Nikah_ceremony, Brinke_Stevens, Yakama_Tribal_Council, Capuchin_Father, wife_Callista_Bisek, Beau_Dare, Bedoni, Arjun_Punj, JOHNNY_KNOXVILLE, cap_tain, Alderwood_Boys, Chi_Eta_Phi, ringleader_Charles_Graner, Savoies, Lalla_Salma, Mrs._Potiphar, fahn, name_Taylor_Sumers, Vernita_Green, Bollywood_baddie, BENBROOK_Texas, Assemblyman_Lou_Papan, virgin_brides, Cho_Eun, CATHY_Freeman, Uncle_Saul, Lao_Brewery, Ibo_tribe, ruf, rival_Edurne_Pasaban, Hei_Shangri_La, Mommy_dearest, interest_Angola_Sonogal, Ger_Monsun, PUSSYCAT_DOLL, Crown_Jewels_Condoms, Lord_Marke, Patootie, Nora_Bey, huntin_shootin, Minister_Raymond_Tshibanda, La_Nina_la_NEEN, signature_Whoppers, estranged_hubby_Kevin_Federline, ”UR”, pill_poppin, ”GEHR”, purebred_Arabians, husbandly_duties, VIAGRA_TIMING, Hereford_heifer, hushed_monotone_voice, Pola_Uddin, Wee_Jimmy_Krankie, Kwakwanso, Our_Galvinator, shoh, Codependency_Anonymous_Group, ”LA”, ”Taufaahau”, Invincible_Spirit_colt, ”SAH_dur”, MOUNT_CARMEL_Pa., watches_attentively, SNL_spinoffs, Seth_Nitschke, Duns_Berwickshire, defendant_Colleen_LaRose, ”Silky_OSullivan”, Highcliff_Farm, ”REN”, Comestar, Satisfied_Frog, Jai_Maharashtra, ATTICA_Ind., lover_Larry_Birkhead, Tami_Megal, chauvinist_pigs, Phi_sorority, Micronesian_immigrant, Lia_Boldt, Sugar_Tits, actress_Kathy_Najimy, zhoo, Colombo_underboss, Katsav_accusers, Bess_Houdini, rap_mogul_Diddy, companions_Khin_Khin, Van_Het, Mastoi_tribe, VITALY, ROLLING_STONES_rocker, womanizing_cad, LILY_COLE, paternal_grandfathers, Lt._Col._Kurt_Kosmatka, Kasseem_Jr., Ji_Ji, Wilburforce, VIAGRA_DOSE, English_Sheepdogs, pronounced_Kah, Htet_Htet_Oo, Brisk_Breeze, Eau_du, BY_MELANIE_EVANS, Neovasc_Medical, British_funnyman_RICKY, 4YO_mare, Hemaida, MONKTON, Mrs_Mujuru, BaGhana_BaGhana, Shaaban_Abdel_Rahim, Edward_Jazlowiecki_lawyer, Ajman_Stud, manly_pharaoh_even, Serra_Madeira_Islands, ”FRAY”, panto_dames, Khin_Myo, dancer_Karima_El_Mahroug, CROWN_Princess, Baseball_HOFer, Hasta_la_Pasta, GIRLS_NEXT_DOOR, Benedict_Groeschel, Bousamra, Ruby_Rubacuori_Ruby, Monde_Bleu, Un_homme_qui, Taylor_Sumers, Rapper_EMINEM, Joe_Menchetti, ”VAY”, supermodel_NAOMI_CAMPBELL, Supermodel_GISELE_BUNDCHEN, Au_Lait, Radar_Installed, THOMAS_TOWNSHIP_Mich., Rafinesque, Herman_Weinrich, Abraxas_Antelope, raspy_voiced_rocker, Manurewa_Cosmopolitan_Club, Paraone, THE_LEOPARD, Boy_Incorporated_LZB, Dansili_filly, Lumpy_Rutherford, unwedded_bliss, Bhavna_Sharma, Scarvagh, en_flagrante, Mottu_Maid, Dowager_Queen, NEEN, model_Monika_Zsibrita, ROSIE_PEREZ, Mattock_Ranger, Valorous, Surpreme, Marwari_businessmen, Grandparents_aunts, Kimberley_Vlaeminck, Lyn_Treece_Boys, PDX_Update, Virsa_Punjab, eyelash_fluttering, Pi_fraternity, HUNTLEIGH_Mo., novelist_Jilly_Cooper, Naha_Shuri_temple, Yasmine_Al_Massri, Mu_Gamma_Xi, Mica_Ertegun, Ocleppo, VIAGRA_CONTRAINDICATIONS, daughter_PEACHES, trainer_Geoff_Wragg, OVERNIGHT_DELIVERY, Fitts_retiree, de_Tourvel, Lil_Lad, north_easterner, Aol_Weird_News, Somewhat_improbably, Sikh_panth, Worcester_2m_7f, Zainab_Jah, OLYMPIC_medalist, Enoch_Petrucelly, collie_Lassie, ”LOW”, clumsiness_Holloway, ayr, ”OHR”, ROLLING_STONES_guitarist, ”LAH_nee”, Ian_Beefy_Botham, Awapuni_trainer, Glamorous_Granny, Chiang_Ching, MidAtlantic_Cardiovascular_Associates, Yeke, Seaforth_Huron_Expositor, Westley_Cary_Elwes, Cate_Blanchett_Veronica_Guerin, Bellas_Gate, witch_Glinda, wives_mistresses, Woodsville_Walmart, 2YO_colt, Manav_Sushant_Singh, Pupi_Avati_Il, Sigma_Beta_Rho, Bishop_Christopher_Senyonjo, Vodou_priest, Rubel_Chowdhury, Claddagh_Ring, ”TAH_duh_al”, ”al_Sadr_mook_TAH”, ROBIN_GIBB, ”GAHN”, BY_THOMAS_RANSON, sister_Carine_Jena, Lyphard_mare, summa_cum, Semenya_grandmother_Maputhi, Clare_Nuns, Talac, sex_hormones_androgens, majeste, Saint_Ballado_mare, Carrie_Huchel, Mae_Dok, wife_Dieula, Earnest_Sirls, spoof_bar_mitzvah, von_Boetticher, Audwin_ Mosby, Case_presentationWe, Vincent_Papandrea, ”KRAY”, Sergi_Benavent, Le_Poisson, Von_Cramm, Patti_Mell, Raymi_Coya, Benjamin_BeBe_Winans, Nana_Akosua, Auld_Acquaintance, Desire_Burunga, Company_Wrangler_Nestea, ask_Krisy_Plourde, JUANITA_BYNUM, livia, GAMB, Gail_Rosario_Dawson, Ramgarhia_Sikh, Catholic_nun_Sister, FOUR_WEDDINGS_AND, Robyn_Scherer, brother_King_Athelstan, Santo_Loquasto_Fences, Wee_Frees, MARISOL, Soliloquy_Stakes, Whatever_Spoetzl, ”MarcAurelio”, mon_petit, Sabbar_al_Mashhadani, ”KAY_lee”, ”m_zah_MAH”, BY_TAMI_ALTHOFF, hobbit_Samwise_Gamgee, Bahiya_Hariri_sister, daddy_Larry_Birkhead, Sow_Tracey_Ullman, coach_Viljo_Nousiainen, Carmen_Lebbos, conjoined_twins_Zainab, Rob_Komosa, ample_bosomed, Ageing_rocker, psychic_Oda]