Alma Mater Studiorum - Università di Bologna

DEPARTMENT OF STATISTICAL SCIENCES
Second Cycle Degree in Statistical Sciences

Accountability in Machine Learning: Comparing Methods for Mitigating Gender Bias in Word Embedding

Presented by:
Clara Biagi

Supervisor:
Prof. Elisabetta Ronchieri

I Session
Accademic Year 2021-2022

See no gender, hear no gender, speak no gender -
see only human, hear only human, speak only human.

Abhijit Naskar, See No Gender

Abstract

Word embedding is widely used in Natural Language Processing tasks. By converting words into numeric vectors, word embedding enables the machine to understand and manipulate text. The models used to create the word vector representation learn all from the text being used for training, including stereotypes and inequalities. It is important that such distortions are not propagated in natural language processing models so that predictions and classifications are fair and unbiased. This work analyses the presence of a gender bias in word embeddings. Considering different methods to remove such bias, the goal is to understand whether there are better methods than others and whether it is possible to completely remove gender bias from the GloVe word embedding.

Contents

Introduction

Artificial intelligence systems are becoming more and more present in our lives every day. They are entrusted with both small daily tasks and important health and safety issues. It is critical that artificial intelligence systems that make decisions, particularly when used in the public sphere, be reliable and accountable. However, this is not always the case and some of these systems have been proven to contain various type of biases. This thesis focuses on one kind of biased system, considering in particular gender bias in word embedding. Word embedding is a vector representation of words, and it is a key component of all natural language processing algorithms. This means that a bias in word embedding will also be present in the natural language processing algorithm that makes use of it, turning out in a final prediction or classification which in turn contains bias. Many methods for removing or reducing bias in word embedding have been proposed. In this work, some of this methods are analysed and compared with the objective of answering at two main questions: is there a best method among them? Is there a method that truly removes bias from word embedding? The analysis focuses on traditional word embeddings omitting the contextualized word embedding.

The thesis is organised in three chapters.

The first chapter introduces the concepts behind the thesis. In the first section, accountability is introduced. Natural language and word embedding are briefly summarised in the second and third sections and then the algorithms for computing some word embedding are explained. The first chapter also introduces the gender bias's problem in natural language processing.

The second chapter focuses more specifically on word embedding. The problem of gender bias in word embedding is explained with the help of some examples. The second section contains a brief summary of related works on the topic. Then, seven different methods for reducing gender bias from traditional word embedding are explained. The last section reports some methods used for evaluating the goodness of word embedding and for assessing the level of gender bias they contain.

The methods and measures introduced in the second chapter are used in the third and last chapter, that contains the comparative analysis. This part represents the core of the thesis. The results reported in this section enable the research questions originally posed to be answered.

The diagram on the following page shows schematically the methodology followed in the structure of this thesis.

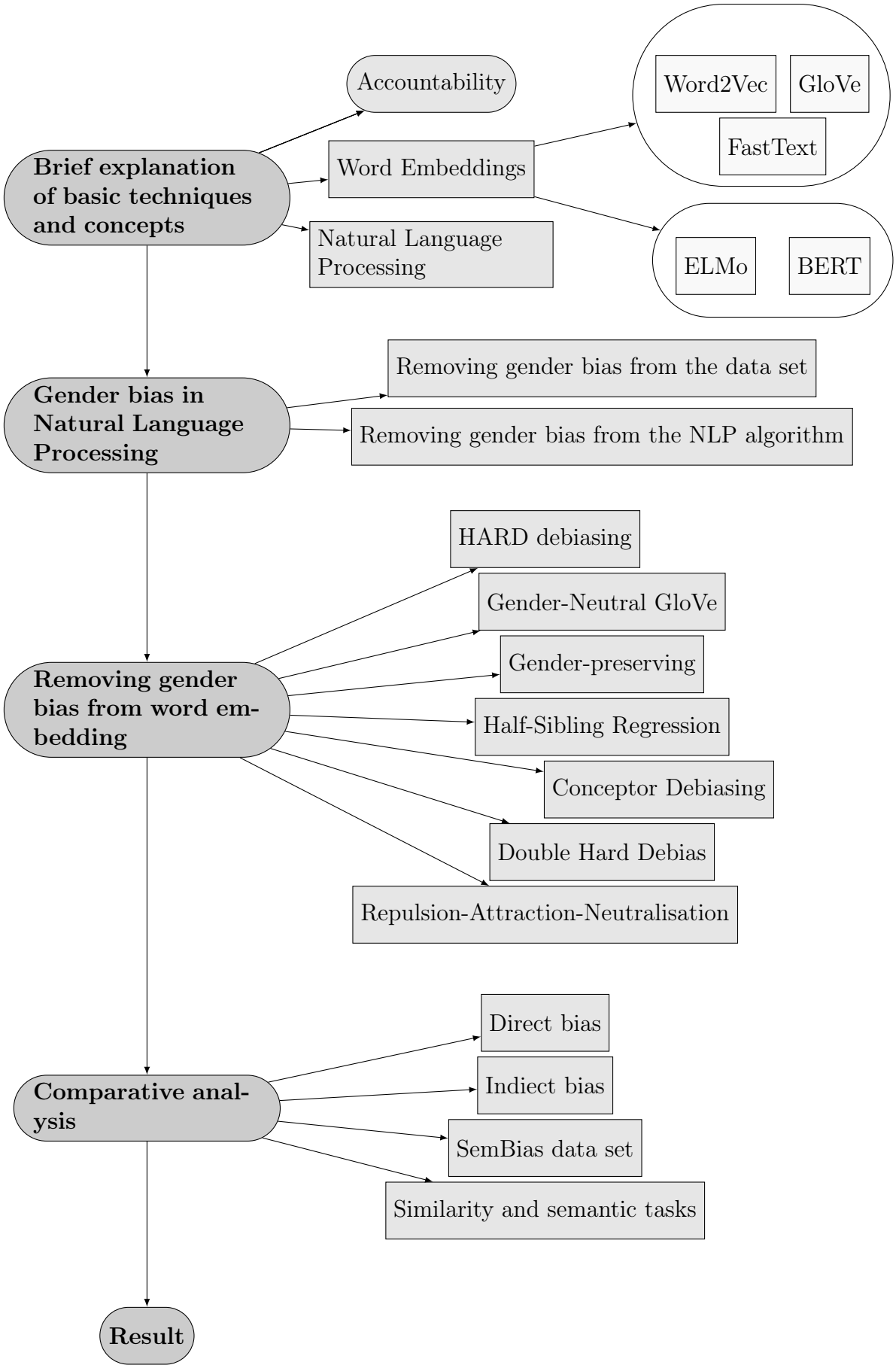


Figure 1: Methodology

Chapter 1

Background

In this chapter basic concepts and methods used in the thesis are introduced. First, the issue of accountability in machine Learning is introduced. Then, one introduces natural language processing (NLP), together with its main applications and techniques. The third paragraph focuses on word embedding. One briefly introduces some embeddings (i.e., word2vec, GloVe, FestText, ELMo, and BERT). Furthermore, the problem of gender bias in NLP with the above-mentioned approaches for addressing it is presented.

1.1 Accountability

The issue of accountability and fairness in machine learning models has raised a lot of concern in recent years. This is a consequence of a wide usage of artificial intelligence (AI) systems in our society, that affects both private and public area. Finance, health care, agriculture and many more fields are already making use of AI to get more efficient and quicker solutions. Great achievements have been made: AI systems help in disease identification and drug manufacturing; they are used to build self-driven cars and facilitate small daily tasks. However, along with many successes come many failures, and whether errors made in contexts that are not particularly risky, e.g., facial recognition in smartphones, are to be fixed but are not particularly harmful, errors made in high-risk problems can be dangerous. In the United States of America, there have been three cases of wrongful arrests based on facial recognition (i.e. Robert Williams in 2020, Michael Oliver and Nijeer Parks in 2019 [17]). AI, used in NLP and facial recognition, has proven to perpetuate societal discrimination and stereotypes [25] [24] [5] [38] [1]. It is not a case that the three men misidentified were all black: ethnic minorities and women are the most prone to errors. Black women in particular have the lowest accuracy in face recognition systems. Another example of artificial intelligence system that may face some ethic problems is the new DALL·E. This model, developed by openAI in 2022, creates images starting from a natural language text. An article by Wired of May 5 [21] already reported some examples of gender and race bias found in DALL·E2, such as that *"eight out of eight attempts to generate images with words like 'a man sitting in a prison cell' or 'a photo of an angry man' returned images of men of color."* These are two examples of many more failures that machine learning and deep learning models have faced. Such errors should be prevented as much as possible, and this

is where accountability is needed.

Bovens [6] defined accountability as "*a relationship between an actor and a forum, in which the actor has an obligation to explain and to justify his or her conduct, the forum can pose questions and pass judgment, and the actor may face consequences*". The AI systems, in order to be accountable, should be able to clearly explain the decision it makes. With this definition Bovens in 2007 defined fundamental steps in the process of accountability: *information*, *explanation*, and in case *consequences*. As outlined by Busuioc [7], the information step faces challenges, such as the problem of opaqueness of black-box models, the privacy of many algorithms which are not available, and the difficulty in understanding complex models. The explanation phase is about the decisions that have been made in the creation of the model. Algorithms have trade-off: *"depending on the value or notion prioritized, the algorithm will reach different results and impose costs and benefits on different individuals or societal groups"* [7]. Knowing the *political* choices at the origin of the implementation is fundamental for accountability. Understanding the influence that algorithms have on human is also part of the explanation phase. Do algorithmic assessments increase the objectiveness of human decisions or do they induce bias instead? May an overreliance on AI could lead us to reduce questioning of outcomes proposed by the machine?

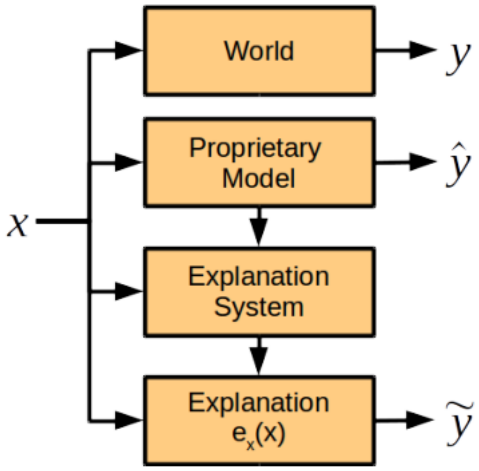


Figure 1.1: Explainable AI system, source [12].

Doshi-Velez et al. [12] focused particularly on improving accountability through explanation. Firstly, they differentiate explanation, which answers at how certain decisions are taken, from transparency, which instead aims at understanding how a model is built. Explanation should shed light on which are the decisive variables in the decision-making process. Having a clear explanation allows identifying and correcting mistakes. Then, Doshi-Velez et al. outlined that AI systems do not interpret data as human do, as they manipulate many variables without knowing what they represent. However, it should be possible to make the system itself provide the necessary explanations, creating an explainable AI system. Its structure is shown in Figure 1.1. The idea is for an overall system composed of two smaller ones: the usual AI system, which tries to predict the true value y with $\hat{y}$ given some data x,

and the explanation system, that give another prediction $\tilde{y}$ starting from the same data x. Explanation system find the prediction through a rule $e_x(x)$ that can be interpreted by human. Comparing $\hat{y}$ and $\tilde{y}$ allows for checking whether the same output is given under the same input data.

Kim and Doshi-Velez [3] identified five key elements for assessing the objectives of AI systems:

- **Transparency** in the selection of data, choice of models and availability of code;
- Usage of **interpretable models** when it is possible, being able to understand why a system takes one decision instead of another, and making it easier to trust in its choices;
- **Post hoc inspection of model outputs**, such as visualizations, statistical tests and sensitivity tests, to evaluate the outcome of the AI system;
- **Empirical performance** quantifies goals, such as fairness and safety;
- Models usually have **properties guaranteed by design**, that usually characterized models and that can help in improving the overall accountability.

Hutchinson et al. [19] focused the attention on the creation of the data set. Errors and biases in the starting data will be reflected in the final model and often the building phase of the data, starting from the original source to metadata and documentation, is not transparent as it should be. Having a clear and clean data set is for sure the first step towards accountability. In the paper the authors claimed that "*the development of machine learning (ML) datasets should embrace engineering best practices around visibility and ownership, as a necessary (but not sufficient) requirement for accountability, and as a prerequisite for mitigating harmful impacts*". Building a data set consists of well defined steps that include asking whether the data are appropriate and how they will be used.

1.2 Natural Language Processing

Natural language processing is the art of applying software algorithms to human language for analyzing, understanding and deriving information from the text data in a smart and efficient manner. It is a sub-field of Artificial Intelligence that connects computer science and linguistics. NLP is the driving technique of many machine learning applications that we are likely to encounter every day, with e.g. google translate, personal assistants like Siri and Alexa and spam detection. Texts are analyzed in their lexicon, syntax and semantics point of view. Text is the most unstructured form of all the available data and it contains various types of noise, so all type of text have to pass a *processing* phase to be analyzable, in order to be understood by the machine. NLP tasks can be computed with traditional machine learning tools and also with neural network architectures.

The data set used for NLP is a *corpus* with its metadata. Source of text could be for example web pages, comments on social media or reviews. A corpus is composed by *tokens*, which are basically the words in it. All unique words, called *type*, create

a vocabulary. There are both *content words* and *stop words*, where the former are words with a semantic meaning, while the latter are words like articles and prepositions. Text processing is the process of cleaning and standardise text, making it noise-free and ready for analysis. It is composed by different steps. A text can be broken into tokens, through the *tokenization*. Words can be reduced to their root or common form with *lemmatization* and *stemming*. With the *part of speech (POS)*, words are marked with their corresponding part of speech tag (noun, verb, etc ...).

Natural language processing is made up by many tasks, which analyse different aspect of texts and language. Examples of tasks are coreference resolution and text classification. The aim of text classification is classifying a text in a specific category. One of its main applications is sentiment analysis, in which a text, e.g. a review, is classified as positive or negative, depending on the emotional tone it conveys. Text classification can be also used to classify texts based on their language or to detect and filter abusive language. Coreference resolution is used to identify all elements that refers to a given entity in a given text. For example, consider the sentence *"Mary told me she was jealous of her brother, because he can play the piano better", I said*. Coreference resolution identifies to whom the various pronouns refer, that is that *she* and *her* refer to *Mary*, *me* refers to *I* and *he* refers to *her brother*.

1.3 Word Embedding

Computers only understand numbers, not words or sentences and word embedding plays a key role in fixing this issue. It is a deep learning model that takes a corpus as input and, while pursuing some particular objective, it produces a list of vectors representing the words in the initial corpus. The resulting vectors reflect both semantic and syntactic meanings and they are defined such that the cosine similarity between vectors indicates the level of similarity between the words represented by those vectors.

Let $X_1 \in \mathbb{R}^d, X_2 \in \mathbb{R}^d$ be *d-dimensional* vectors. The cosine similarity between X_1 and X_2 is given by

$$cos(X_1, X_2) = \frac{X_1 \cdot X_2}{\|X_1\|\|X_2\|} \tag{1.1}$$

We can use arithmetic of the embedding vectors to obtain meaningful relations, e.g. $\overrightarrow{Italy} - \overrightarrow{Rome} + \overrightarrow{Paris} = \overrightarrow{France}$.

Although they can be trained as needed, pretrained word embedding are available and they are the ones most commonly used, given that they are computationally expensive.

1.3.1 word2vec

An important word embedding, known as **word2vec**, was proposed by Mikolov et al. [32]. They presented two different neural network architectures for computing vector representation of words: the Continuous Bag-of-Words model (CBOW) and the Continuous Skip-gram model (see Figure 1.2).

The CBOW model consists of input, projection and output layers, and it tries to predict a given word based on the context - i.e., future and history words. The number of considered context words depends on a window size: e.g. a window size of k means that $k/2$ words before and $k/2$ after the target word are considered. The Skip-gram model is similar to the CBOW, but it reverses input and output: it takes as input a single word and predicts words within a certain range before and after the current word. The main difference identified by the authors is that CBOW is faster while Skip-gram is better for infrequent words. Going further into details, word2vec works in the following way: it starts from a corpus and selects from it the target words; then, with a given window size, it selects a set of contextual words for each target word. The target and the contextual words create the training set for a one-layer neural network. Depending on using the Continuous Bag-of-Words or the Skip-gram architecture there will be opposite inputs and outputs. As the model improves, through back propagation and gradient descent, we get the weight of the hidden layer. Those weight represents the final vector representation of the target word.

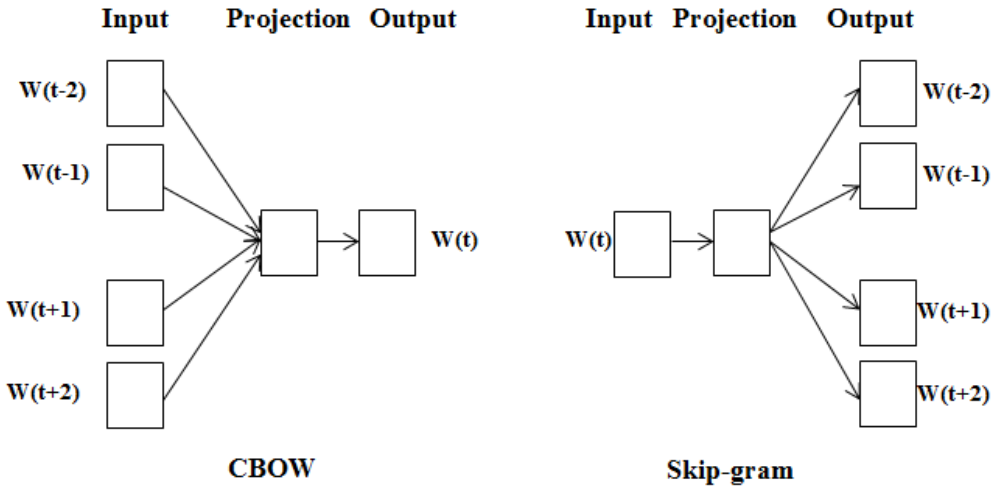


Figure 1.2: CBOW and Skip-gram models architecture.
Source by Mikolov et al. [32].

1.3.2 GloVe

Pennington et al. [33] introduce a different word embedding, called **GloVe** that stands for Global Vector. Words vector in this case are found by looking at the statistics of word occurrences in a corpus. One important difference between word2vec and GloVe is that the former is a predictive model while the latter is a count-based model. The GloVe representation is based on the co-occurrence matrix schematized in Table 3.1: the entry X_{ij} of this matrix is the number of times the word j occurs in the context of the word i.

	$word_1$	$word_2$	...	$word_i$	...	$word_V$
$word_1$	0	X_{12}	...	X_{1i}	...	X_{1V}
$word_2$	X_{21}	0	...	X_{2i}	...	X_{2V}
...	...	...	...	...	...	...
$word_j$	X_{j1}	X_{j2}	...	X_{ji}	...	X_{jV}
...	...	...	...	...	...	...
$word_V$	X_{V1}	X_{V2}	...	X_{Vi}	...	0

Table 1.1: Co-occurrence matrix.

Using the co-occurrence matrix, one can define the probability that the word i occurs in the context of word j as $P(i|j) = \frac{X_ij}{\sum_{k=1}^{V} X_{ik}}$. A comparison between words can be performed through ratios of probabilities:

$$\frac{P(i|k)}{P(j|k)} \tag{1.2}$$

When the ratio expressed in Equation 1.2) is greater than 1, the word i occurs more often than the word j in the context of the word k; when the ratio is under 1, the word j occurs more often than the word i in the context of the word k; instead, when it is almost 1, the word i and the word j are both related or unrelated to k. In real application, building the matrix of co-occurrence can be challenging because of the curse of dimensionality. As a consequence, Equation 1.2 is replaced by its estimation:

$$F(w_i, w_j, \tilde{w_k}) = \frac{P(i|k)}{P(j|k)}, \tag{1.3}$$

where w_i and w_j are the vectors of word $i-th$ and $j-th$ and $\tilde{w_k}$ is the vector of the context word. Considering the vector difference and then using the dot product bring to $F((w_i - w_j)^T \tilde{w_k})$, which contains the information in Equation 1.2. Now, function F has to be chosen. To do that, it can be considered that F is a homo-morphism, which implies that

$$F((w_i - w_j)^T \tilde{w_k}) = \frac{F(w_i^T \tilde{w_k})}{F(w_j^T \tilde{w_k})} \tag{1.4}$$

but

$$F((w_i - w_j)^T \tilde{w_k}) = \frac{P(i|k)}{P(j|k)} \tag{1.5}$$

from which

$$F(w_i^T \tilde{w_k}) = P(i|k) = \frac{X_{ik}}{\sum_{k=1}^{V} X_{ik}} \tag{1.6}$$

Now, the function only depends on two words instead of three. Choosing $F = e^A$, the homo-morphic property is respected and one gets

$$w_i^T \tilde{w_k} = \ln P(i|k) = \ln X_{iK} - \ln \sum_{k=1}^{V} X_{ik} \tag{1.7}$$

A bias b_i for w_i and a bias $\tilde{b_k}$ for $\tilde{w_k}$ should be added. Given that $\sum_{k=1}^{V} X_{ik}$ does not depend on k, it can be incorporated in b_i.

$$w_i^T \tilde{w_k} + b_i + \tilde{b_k} = \ln X_{ik} \tag{1.8}$$

One important drawback of this equation is that it weight all co-occurrence equally, while rare co-occurrence brings less information and should be considered less in the construction on the word vectors. Then, the final solution is to consider a weighted least square regression model:

$$\sum_{i,j=1}^{V} f(X_{i_J})(w_i^T \tilde{w_j} + b_i + \tilde{b_j} - \ln X_{i_j})^2 \tag{1.9}$$

where f is a generic function. Minimizing this loss function for the word vectors one obtains the GloVe representation.

1.3.3 FastText

Bojanowski et al. [4] present an improvement to the word2vec word embedding. The resulting methods is fast and allows to learn quickly from large corpora and to compute word vector for words that lack in the training data. This new model, called **FastText**, was designed to capture the morphology of rich languages by considering the internal structure of words. Both Word2vec and GloVe represent each word in the vocabulary by a distinct vector, and they do not share any parameter. Starting from the Skip-gram architecture, the FastText word embedding models morphology by considering sub-word units. Words are then represented by a sum of its character n-grams. N-grams are continuous sequences of words or symbols or tokens in a document. For example, considering *n=3*, the vector representation for the words *dancer* is found by summing its character n-grams, which are *dan*, *anc*, *nce*, *cer*.

1.3.4 ELMo

ELMo (Ebedding from Language Models) [34] is a deep contextualized word representation. As previous embeddings, it models complex characteristics of words like syntax and semantics. However, unlike other methods, it menages to find different embeddings for the same word when it has different meanings: it can handle polysemy by considering the context.

The contextual word vectors are obtained by the combination of the internal states of a pre-trained bidirectional Long Short Term Memory Language Model. The first layer of the model captures the syntax part of a word while the high-level LSTM layer summarises more context-dependent aspect: ELMo representation is called *deep* precisely because the embeddings are functions of all of the internal layers of the biLM. LSTM-LM is simply a Long Short Term Memory neural network trained with a language model objective. *Bidirectional* because it combines both a forward and a backward language model. Briefly, a forward language model takes a sequence of N tokens as input and computes the probability of the token t_k given the previous tokens $t_1, \cdots, t_{k-1}$. A backward language model takes the same

sequence as input and computes the probability of the token t_k given the future tokens $t_{k+1}, \cdots, t_N$. The bidirectional mechanism pushes vectors to depend on the entire sentences, allowing words to have different embeddings in different context. ELMo's architecture is shown in Figure 1.3.

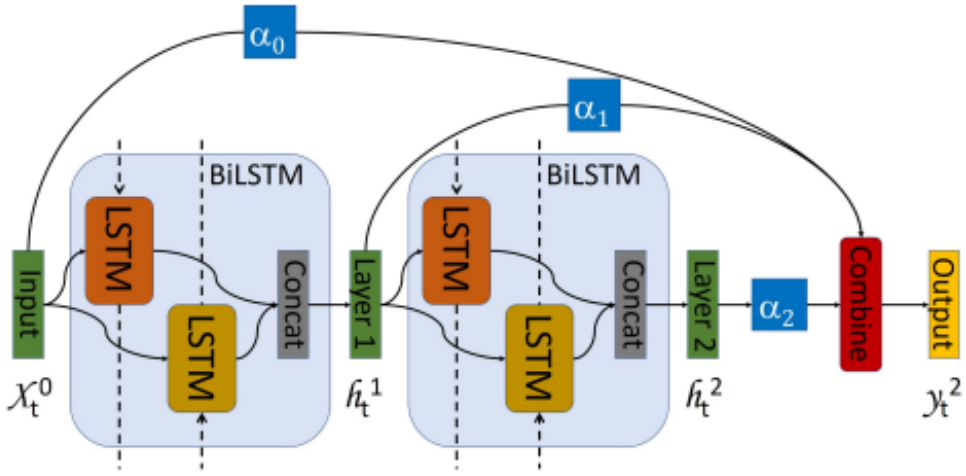


Figure 1.3: Structure of neutral network that produces the ELMo embedding. Source [26]

Starting from a sentence, tokens are separated and a context-independent representation is obtained through a character level CNN (x_t^0 in 1.3). Creating a character level word embedding allows to take into account the morphological aspect of words. This raw word vectors are the input of a 2-layer bidirectional LSTM-LM. Backward and forward components generate 2 intermediate word vectors, which are then concatenated to each other and put as input for the second LSTM layer. This layer has identical structure to the firs and it outputs 2 others intermediate word vectors which are in turn concatenated together. Eventually, ELMo word embeddings are a weighted combination of 3 representations: the context-independent and the 2 concatenation of couples of backward and forward intermediate representation.

1.3.5 BERT

BERT is the state-of-the-art method to extract vectors from text data and it is a contextual word embedding like ELMo. It was introduced by Jacob Devlin et al. [9]. The name BERT comes from *Bidirectional Encoder Representation from Transformer*. It is indeed based on architecture of the Transformer (see Figure 1.4).

Transformers are deep learning models built for machine translation introduced in 2017 by Google [39]. They are composed by an encoder and a decoder and they both contain attention mechanism (i.e. the ability to focus on different parts of the input, according to the requirements of the problem being solved) and a feed forward neural network. The encoder and the decoder are themselves composed by a stack of encoders and decoders respectively. The encoder component outputs word embeddings for each words while the decoder outputs the final translation. So, the encoder is the one that learn and understand the language and the context. The

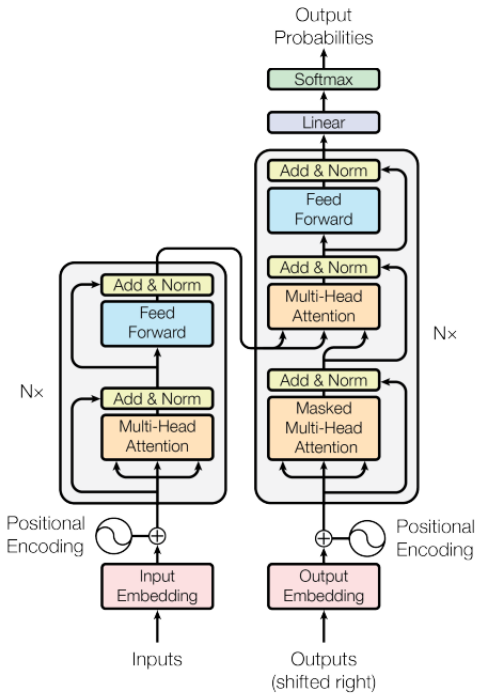


Figure 1.4: Transformer architecture.
Source [39].

important upgrade of transformers over bidirectional LSTM is given by the ability of transformers of processing all words simultaneously, while LSTM network processes one word at the time. This makes transformers faster.

In the original BERT paper, the authors introduce two steps: *pre-training* and *fine-tuning*. The former concerns training the model for multiple tasks and the latter uses pre-trained parameters but is then fine-tuned for specific tasks. The pre-trained phase is the one of interest for understanding how word embedding are computed. During this step, BERT is trained on two unsupervised tasks: masked language model (MLM) and next sentence prediction (NSP). The goal of MLM is to output the correct masked words, starting from a sequence of words where a random portion of them is masked. The prediction is then computed by understanding the context. NSP instead takes two sentences as input and it the objective of understanding if the second sentence follows the first or not. As shown in Figure 1.5, BERT takes two sequences as input, *masked sentence A* and *masked sentence B*. As first step, all tokens are converted into vector representation, which is obtained by a combination of three components as shown in Figure 1.6: token embedding, segment embedding and position embedding. The token embedding is a pretrained embedding (in the paper the WordPiece is used), segment embedding indicates if the token belongs to sentence A or B and position embedding represents the position of the token in the sequence it belongs. Information about the token position are needed because sentences will be considered simultaneously but the sequential nature must be preserved in order to capture the context and the meaning.

The word vectors are then processed by BERT until they reach the output layer, composed by a binary element for the NSP task and the embeddings for the masked (and also unmasked) words. Each layer of the BERT model return a possible word

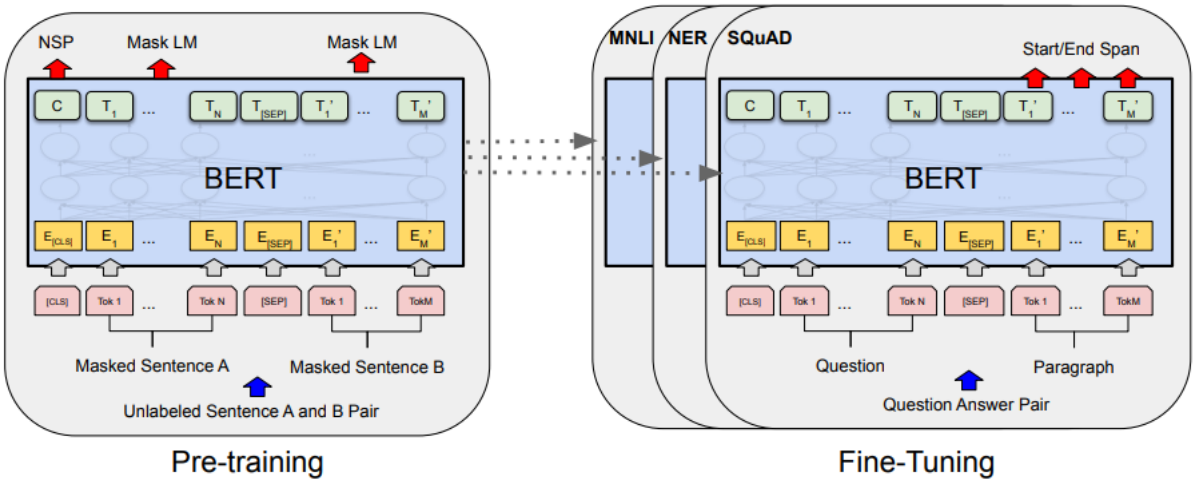


Figure 1.5: BERT architecture.
Source [9].

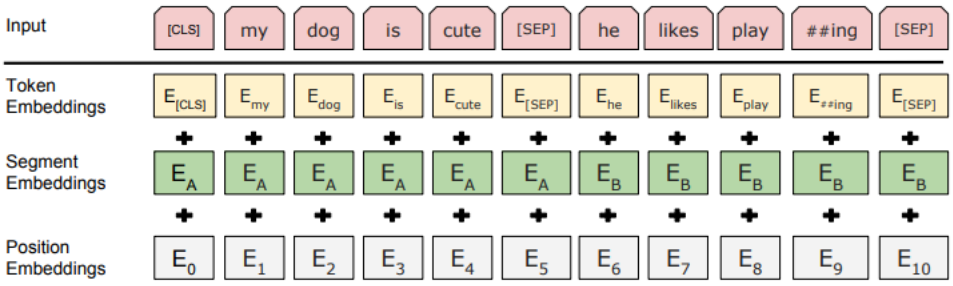


Figure 1.6: How initial word embedding for BERT are computed.
Source [9].

embedding. Authors suggest to concatenate the last 4 layers to obtain the best vector representation. Two different size of BERT are proposed: The base BERT architecture has 12 layers, a hidden size of 768 and 12 self-attention heads, for a total number of parameters of 110M. The large BERT has 24 layers, a hidden size of 1024 and 16 self-attention heads, for an overall number of parameters of 340M. The large BERT reach the best result, but with much more parameters.

1.4 Bias in Natural Language Processing

Human texts are full of noises and language is not always clear. For example, ironic sentences may be very hard to interpret for a machine and depending on tone, equal sentences can have opposite meanings. Not to mention the differences and peculiarities of different languages. Text corpora, unlike numerical data, carries cultural and social meanings that are intrinsic components for a language. Texts referring to different subgroups, relative to age, gender, ethnicity, may have different connotations. Socio-demographic differences introduce bias as they carry information and this is reflected, nor in good or bad sense, into natural language processing algorithms. However, one must be careful that these biases do not penalise one group or another.

Hovy and Prabhumoye [18] summarised five source of bias in natural language that are following reported.

- A first bias comes from the text: using a text for training a machine learning model will result in a fair algorithm for the population the text represents but should not be generalized to others. Having a balanced data set where all subgroups are well represented is a good starting point for accounting for unwanted biases.

- Another bias still related to the collection of the data is the one generated by wrong labels. Bias from labels can arise from inattention of annotator, when more than one label can be chosen or when authors and annotators are characterized by different social norms, which may lead to different interpretations.

- As it will be discussed in the next chapter, word embedding is a source of bias. It captures semantic biases that will be contained in the final predictions.

- The choice of the model can also influence the resulting bias. In particular, using a model trained on biased data may lead to an amplification of that bias when used on new data.

- The last identified source of bias is related to the research design. The authors argued that the most studies concerns Indo-European data. Improvements are mostly made in English models, while studies and papers about Asian and African languages are scarce. This in turn does not encourage studies on the topic and accentuates even more the different level of depth and knowledge between English and other languages.

There are many ways for addressing the problem. Three different kind of approaches are summarized by Sun et al. [37] as follows:

1. working on text corpora
2. working on the word embedding
3. working onto the NLP algorithms adjusting in some way the predictions

Methods for debiasing word embeddings are illustrated in the next chapter. Here some examples of the other two approaches for mitigating the problem of gender bias are briefly presented.

1.4.1 Removing gender bias by modifying the corpora

Natural language systems learn biases from the text corpora used for training. As a consequence, removing gender bias directly from the corpora is probably the more immediate way of solving the problem. One of the first attempts was done by Zhao et al. [43]. Their idea was to augment the original corpus creating a new corpus with all gendered entries swapped: male elements are converted to female and vice-versa. This approach solves the problem of an underrepresented gender, even if it doubles the size of the training data, and successfully decreases gender bias in coreference resolution system. Lu et al. [30] expanded the method by proposing the *counterfactual data augmentation*, that is a generic method for addressing the problem of gender bias in neural natural language processing tasks. Babaeianjelodar et al. [2] used different training corpora with BERT word embedding and measured the gender bias for each of them. They proved that different corpora may lead to different level of gender bias, and that apparently fair corpora like Google News corpus may have a bias even higher than the one derived by evidently stereotypical texts. Working directly on the corpus has the big advantage that the unbiased corpus can then be used for multiple model and aims.

1.4.2 Removing gender bias by modifying the algorithm

Another approach for removing gender bias in natural language processing is by working on a given NLP algorithm. In this way the debiased solution is found also considering peculiar aspects of the specific task but it could hardly be generalized. Example of this solution is adversarial learning [42]. When predicting an outcome Y, the idea is adding an *adversarial network* to mitigate bias. In this model (shown in Figure 1.7) there are two steps: the first concern the prediction of the outcome Y, whose prediction accuracy has to be maximized. In the second, a protected variable Z is predicted starting from $\hat{Y}$, and its accuracy has to be minimised. In the paper, the task of predicting the income of subjects is considered. In this case, once the income has been predicted, it should not be possible to accurately predict the protected variable. The protected variable could be gender, race or any variable that should be unbiased.

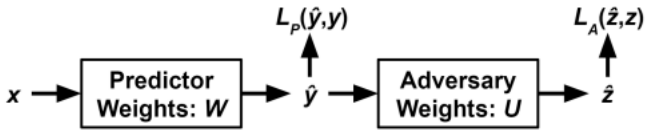


Figure 1.7: Structure of the adversarial learning. Source [42].

In the context of language models, [35] proposed a technique based on the modification of the loss function. Other methods have been proposed for machine translation [13] [36], dialogue generation [10], text classification [11] and for dealing languages with rich morphology [46] or grammatical gender [45].

Chapter 2

Gender bias in word embedding

This section shows that word embeddings are biased towards gender, and a brief summary of previous works on the topic is included. Some methods for reducing gender bias are presented in detail together with some measures or tasks to evaluate the resulting debiased word embeddings.

2.1 Gender Bias in word embedding

As already said, word embedding is widely used in natural language processing. However, one dangerous feature of word embedding is that it learns and amplifies biases present in the text used for training. Corpora reflect constructs, stereotypes and inequalities present in the society that produced them, and when they are used to train word embedding it inherits and learns also sexist and racist associations. This is not a problem itself: Garg et al. for example exploited this property of word embedding to quantify gender and ethnic stereotypes over time in their paper "Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes" [14]. However, when word embedding are used in NLP tasks, having intrinsic biases constitutes a problem. Consider using a word embedding for the task machine translation, what if the word embedding contains gender bias? In the following table are reported some translation from English to Italian that show a gender issue:

English	Italian
plumber	idraulico
programmer	programmatore
professor	professore
chef	cuoco
volunteer	volontario
homemaker	casalinga

Table 2.1: Gender biased translation from `https://www.deepl.com/it/translator` and from `https://www.wordreference.com/it/`

All these words in English are the same in masculine and feminine, while Italian has a different ending for the two genders. It is evident from the translation that some words are translated into masculine and some into feminine according to stereotypes:

for these translators, *plumber* is more masculine and *homemaker* is more feminine. A fairer translation is given by google translate as follows:

English	Italian	
plumber	idraulica (*female*)	idraulico (*male*)
programmer	programmatrice (*female*)	programmatore (*male*)
professor	professoressa (*female*)	professore (*male*)
chef	capocuoca (*female*)	capocuoco (*male*)
volunteer	volontaria (*female*)	volontario (*male*)
homemaker	casalinga (*female*)	casalingo (*male*)

Table 2.2: Gender unbiased translation from `https://translate.google.it/?hl=it`.

Other examples of the problematic usage of a biased word embedding are resume filtering system and the task of putting in order relevant web pages in web engine. Using an unfair word embedding, as outlined by Bolukbasi et al. [5] may lead to an amplification of already present biases. The problem is that if the word embedding has learnt the association between *woman* and *homemaker* and between *man* and *computer programmer*, when searching for *computer programmer* in a search engine or when filtering a resume for finding a programmer employee, all pages and resume related to women will be discarded. This mechanism will amplify the social bias instead of reducing it. Similar biases appear in word embeddings for other human stereotypes like race [31]. As outlined previously, the cosine similarity between vectors measures the level of similarity between the words represented by them. Here are reported some analogies detected by the GloVe embedding that show a gender bias.

she	seamstress	he	**bricklayer**
he	military	she	**civilian**
man	woman	programmer	**suffragist**
he	strong	she	**weak**

Table 2.3: Biased analogies found in GloVe word embedding.

Trying to complete the analogy "man is to computer programmer as woman is to x" with Word2Vec word embedding turned out that the solution was "homemaker". Why is this the case? The best guess for compiling the analogy is found through simple arithmetic of the embedding vectors. The word "homemaker" is found because $\overrightarrow{man} - \overrightarrow{woman} \approx \overrightarrow{computer\ programmer} - \overrightarrow{homemaker}$.

This means that in the space of word embedding, words like "homemaker" are more similar (nearer) to "woman" than to "man". But why? That's an easy answer: because it learnt those associations from the corpus it was trained on.

2.2 Summary

The problem of bias in machine learning has already been explored by some authors.

Bolukbasi et al. [5] first showed that word embedding contain biases. They discovered that stereotypical and discriminating answers were given by the word embedding when used for solving analogy tasks. They proposed a way of quantifying bias in word embdedding, creating a distinction between a direct and an indirect bias, and two methods to reduce it (hard and soft debiasing methods). They used Word2Vec as word embedding, trained on 3 million English words from Google news. Their technique is based on the projection of the biased word vector onto a space that is orthogonal to the gender subspace.

Another debiasing method was proposed by [44]. One important difference between this method and the previous is that in this case gender-neutral words are jointly identified while learning word vectors, while in hard and soft debiasing methods there were an additional classifier to use. Authors removed gender bias from word embedding by modifying the loss function of GloVe.

In 2019, Gonen and Goldberg [15] showed how the two previous methods succeeded in removing direct bias but did not manage to fix indirect bias. As result, they only hide gender bias while not really removing it. In the paper they proposed 5 tasks through which one can identify the remaining indirect gender bias in a words embedding.

Yang et al. [41] reduced gender bias through statistical dependencies. Estimate of gender bias is obtained by predicting neutral words through gender-definition words and then it is subtracted from original word vectors that are to be debiased.

Kaneko and Bollogala [22] minimised the gender bias by optimising a loss that depends on four components: two components controls the feminine/masculine element of words, one controls their neutrality and the last component controls the loss of semantic information.

Conceptor debiasing method introduced by Karve et al. [23] has the peculiarity that it menages to remove multiple biases simultaneusly. It first identifies a biased subspace through a conceptor matrix and then it applies its negation to a given word to shrink its bias.

In 2020, Wang et al. [40] presented an improvement to the hard debiasing method. They realised that word frequency in the corpora can influence the performance of hard-debiasing by twisting the gender direction. Therefore, they first mitigated the impact of word frequencies and then applied hard-debiasing.

Kumar et al. [27] introcuded a method, the RAN-Debias, that reduces bias from a target word by modifying its neighbours representation and not only its direct bias. They introduced also a new measure of bias, the Gender-based Illicit Proximity Estimate (GIPE).

2.3 Methods for reducing bias

2.3.1 Hard and Soft be-diasing methods

To quantify the bias of a specific word Bolukbasi et al. [5] compare it to a couple of gender-specific words (e.g. *he-she*): bias is present if the word, which it is supposed to be neutral, is closer to *she* than *he* or vice-versa.

They identify two kind of bias:

- **Direct bias**, when there is an association between gender neutral words and gender pairs. *Homemaker* is closer to *woman* than *man*,
- **Indirect bias**, when the association is between two neutral words. The fact that *receptionist* is much closer to *softball* than to *football* comes from the association of both *receptionist* and *softball* with female words.

Measure of bias, both direct and indirect, as well as their debiasing method, is based on the definition of a gender-specific direction, which captures much of the gender bias. To identify it, the authors consider 10 gender-pairs and compute the difference between each couple of embedding vectors. Then, they perform principal component analysis on the ten gender pair difference vectors. Once found this direction g (one PC could be enough to capture most of gender information, but more than one PCs can be considered), the direct gender bias of a word embedding is given by

$$\frac{1}{|N|}\sum_{w\in N}|cos(\overrightarrow{w}, g)|^c \tag{2.1}$$

where N are the gender neutral words, $\overrightarrow{w} \in N$, g is the identified gender subspace and c indicates the level of bias we want to detect: when 0, the gender bias is zero only if there is no overlap between word $\overrightarrow{w}$ and g, otherwise *c=1*. However, this measure does not consider *indirect bias*. Indirect bias can be computed considering that a generic word vector $\overrightarrow{w}$ can be decomposed as $\overrightarrow{w} = \overrightarrow{w_g}+\overrightarrow{w_\perp}$, i.e. the projection of the word vector into the gender subspace $\overrightarrow{w_g} = (\overrightarrow{w_g} \cdot g)g$ and the vector without gender component $\overrightarrow{w_\perp} = \overrightarrow{w} - \overrightarrow{w_g}$. Thanks to this decomposition the idea is to quantify the contribution of the gender subspace to the similarity between any couple of words $\overrightarrow{w}, \overrightarrow{v}$ as

$$\beta(\overrightarrow{w}, \overrightarrow{v}) = \frac{(\overrightarrow{w} \cdot \overrightarrow{v} - \frac{\overrightarrow{w}_\perp \cdot \overrightarrow{v}_\perp}{\|\overrightarrow{w}_\perp\|_2 \|\overrightarrow{v}_\perp\|_2})}{\overrightarrow{w} \cdot \overrightarrow{v}} \tag{2.2}$$

which quantify how much the inner product between $\overrightarrow{w}$ and $\overrightarrow{v}$ changes after removing the gender subspace.

The debiasing algorithm consists of two phases:

1. identify gender subspace (or gender direction, if the dimension is 1):
2. neutralize and equalize or soften.

While the first phase is fixed, the second one has two alternatives to choose; according to the choice made, **Hard debiasing** or **Soft bias correction** will be applied.

Hard debiasing forces all gender neutral word to be orthogonal to the gender direction while in the soft method the level of debiasing can be controlled by a tuning parameter.

In the hard debiaseing algorithm, a list of words to neutralize and a family of equality sets is taken as input together with the initial word embedding. The equality set is of the form $\varepsilon = \{E_1, E_2, ..., E_m\}$, where each E_i is a couple of female-man words like {*man, woman*}, {*businessman, businesswoman*}. The algorithm define a new embedding for all words in the original embedding: words to be neutralized will have an embedding given by

$$\overrightarrow{w} = \frac{\overrightarrow{w} - \overrightarrow{w_g}}{\|\overrightarrow{w} - \overrightarrow{w}\|}; \tag{2.3}$$

all words in a specific equality set E instead will have an embedding given by

$$\overrightarrow{w} = \nu + \sqrt{1 - \|\nu\|^2} \frac{\overrightarrow{w_g} - \mu_g}{\|\overrightarrow{w_g} - \mu_g\|}, \tag{2.4}$$

where $\nu = \mu - \mu_g$ is the mean vector orthogonal to the gender direction and $\mu = \sum_{w \in E} \frac{w}{|E|}$ is the mean vector for the equality set E.

The soft bias correction takes as input two matrices: the one of all embedding vectors $W \in \mathcal{R}^{d \times |vocab|}$ and the one with the embeddings of only the neutral words N. The output is a linear transformation of the two matrices that minimize the projection of gender neutral words on the gender subspace while maintaining the similarity (i.e. the inner product) between word vectors. This optimization problem is controlled by a tuning parameter λ: the higher λ, the more projection is removed and for λ large soft bias correction gives the same result has hard debiasing, by removing the gender projection by all the gender neutral words.

Gender neutral words are found by subtracting the gender specific words from the whole list. They first create a list of 218 gender specific words looking into only 26377 words out of the entire 3 million words of the word2vec embedding and then they generalize the list using a linear Support Vector Machine. The main drawback of this method is that finding gender neutral words in this way add a source of error given that the SVM classifier can make mistakes.

Figure 2.1 shows a diagram with all the steps needed in order to debias a word embedding following Bolukbasi's procedure.

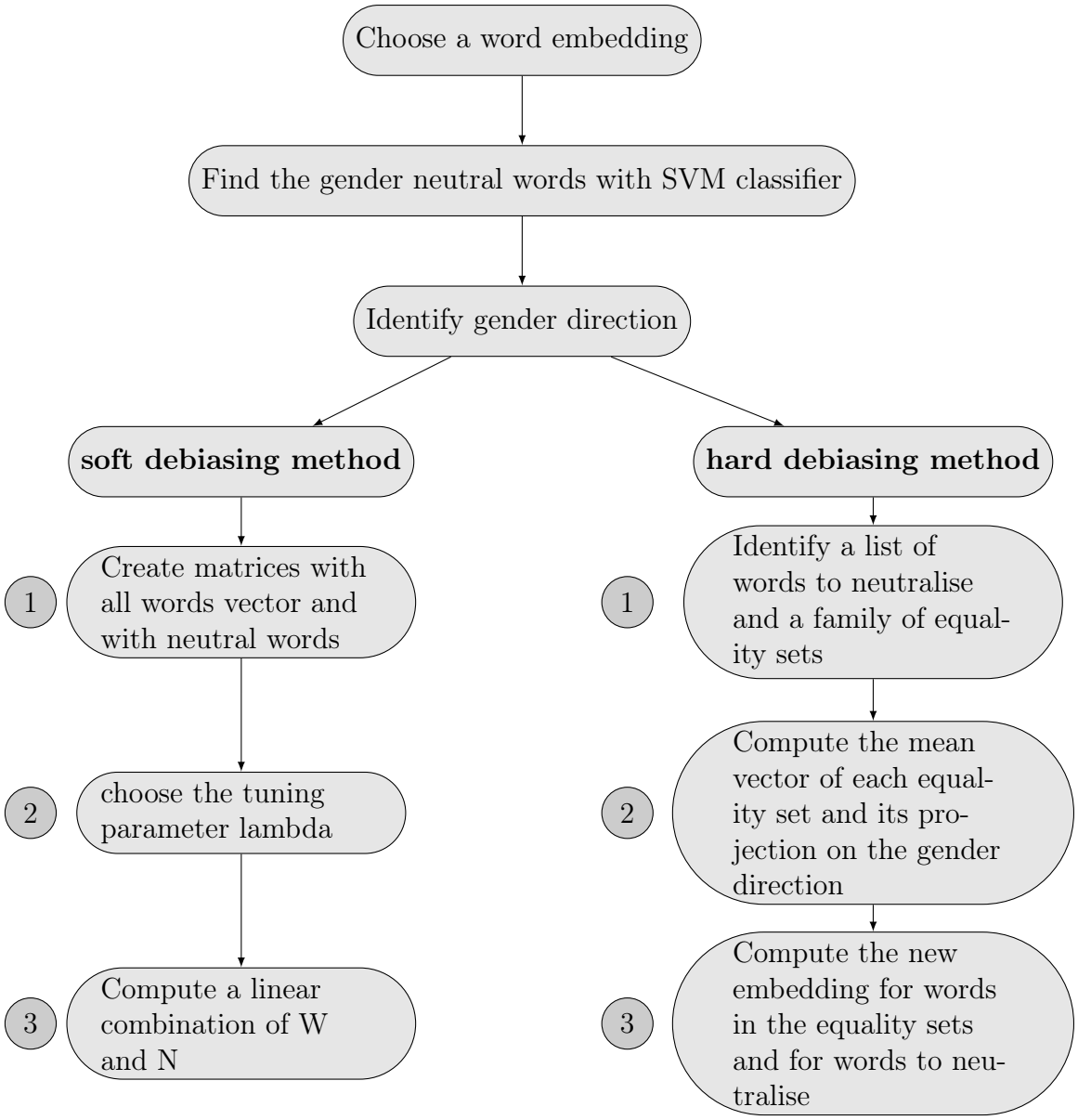


Figure 2.1: Steps of the Hard and Soft debiasing methods

2.3.2 Gender-Neutral GloVe (GN-GloVe)

Zhao et al. [44] proposed a method that decrease gender bias while training the word embedding, instead of correcting a pre-trained one. They used the GloVe representation and found word vectors through a modification of the loss function of GloVe. The key idea is that each word w in the embedding consists of a gendered component $w^{(g)}$ and a neutralized component $w^{(a)}$: the method aims at keeping all the gender information into one component while making the other independent of gender influence. The new vectors will have the gender component concentrated in the last coordinate, and it may be kept or not. Word vectors are found by minimizing the following expression

$$J = J_G + \lambda_d J_D + \lambda_e J_E \tag{2.5}$$

J_G is the loss function from the GloVe embdedding (Equation 1.9).
J_D aims at restricting gender information into $w^{(g)}$ and it has two different formulation:

$$J_D^{L1} = -\| \sum_{w \in \Omega_M} w^{(g)} - \sum_{w \in \Omega_F} w^{(g)} \|_1 \tag{2.6}$$

$$J_D^{L2} = \sum_{w \in \Omega_M} \|\beta_1 e - w^{(g)}\|_2^2 + \sum_{w \in \Omega_F} \|\beta_2 e - w^{(g)}\|_2^2 \tag{2.7}$$

Equation 2.6 minimizes the negative distance between male words Ω_M and female words Ω_F. In Equation 2.7 values of word vectors are restricted in the range $[\beta_1, \beta_2]$, where β_1 and β_2 are arbitrary number, chosen by the authors to be 1 and -1, and $w^{(g)}$ is pushed into one of the extremes. $e \in R^k$ is a vector of ones.
J_E pushes all neutral words Ω_N to have their $w^{(a)}$ orthogonal to the gender subspace v_g:

$$J_E = \sum_{w \in \Omega_N} (v_g^T w^{(a)})^2 \tag{2.8}$$

The gender direction is found as as

$$v_g = \frac{1}{|\Omega'|} \sum_{(w_m, w_f) \in \Omega'} (w_m^{(a)} - w_f^{(a)}) \tag{2.9}$$

where Ω' is a set of gender word pairs.

λ_d and λ_e in Equation 2.5 are hyperparameters that tune how much to remove the bias from word embedding. The main limitation of this method is that it can be applied only to GloVe embeddings, or more precise to word embedding computed through minimisation of a loss function.

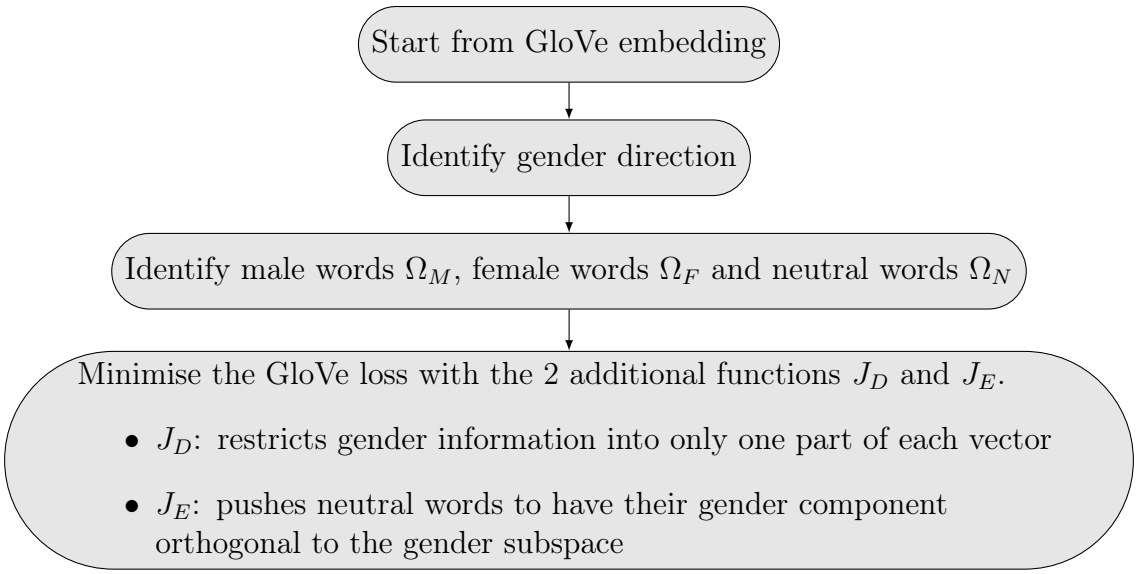


Figure 2.2: Steps of the GN-GloVe method

2.3.3 Gender-Perserving GloVe (GP-GloVe)

Kaneko and Bollogala [22] underlined the importance of removing gender bias while keeping gender orientation that are not unfair. The final purpose is to learn a map $E : R^d \to R^l$ that projects the original pre-trained word embeddings to a debiased l-dimensional space. They started by splitting the vocabulary into four mutually disjoint categories: female oriented V_f, male oriented V_m, neutral V_n and stereotypical V_s. The total vocabulary V is given by the union of these sets. Female and male oriented words are those words like *bear* or *bikini* that have an fair gender bias while the stereotypical words are those like *homemaker* that have a unfair gender bias. The aim of their method is to protect feminine and masculine properties when required, preserve gender gender neutrality when needed and remove gender bias from stereotypical words. Two functions are considered:

$$C_f = R^l \to [0, 1] \tag{2.10}$$

$$C_m = R^l \to [0, 1] \tag{2.11}$$

C_f predicts the degree of femininess and C_m predicts the degree of masculinity of a given word w. 1 is assigned to highly feminine or masculine words respectively. The function E is learnt as the encoder of an autoencoder whose decoder is given by $D : R^l \to R^d$. To protect feminine and masculine aspect of female and male oriented words, the function E should minimise the following losses:

$$L_f = \sum_{w \in V_f} \|C_f(E(w)) - 1\|_2^2 + \sum_{w \in V \setminus V_f} \|C_f(E(w))\|_2^2 \tag{2.12}$$

$$L_m = \sum_{w \in V_m} \|C_m(E(w)) - 1\|_2^2 + \sum_{w \in V \setminus V_m} \|C_m(E(w))\|_2^2 \tag{2.13}$$

The first summation of the equations forces the feminine or masculine component of a female or male oriented word to be close 1 while the second sum forces the gender part to be as small as possible for all words that do not have a fair gender component.
To preserve gender neutrality for gender neutral words and to remove gender bias from stereotypical words, the strategy is to project them into a subspace orthogonal to the gender ν_g. Gender direction is found as in previous methods using a set of feminine and masculine word-pairs. Minimizing the squared inner-product between gender direction and stereotypical or gender words lead to this result:

$$L_g = \sum_{w \in V_n \cup V_s} (\nu_g^T w)^2 \tag{2.14}$$

To avoid losing semantic information encoded in the embedding while debiasing, the reconstruction loss is also minimised:

$$L_r = \sum_{w \in V} \|D(E(w)) - w\|_2^2 \tag{2.15}$$

Eventually, the map E is found by simultaneously minimise all the losses presented above, leading to the final loss function:

$$L = \lambda_f L_f + \lambda_m L_m + \lambda_g L_g + \lambda_r L_r \tag{2.16}$$

where $\lambda_f, \lambda_m, \lambda_g, \lambda_r$ are hyper-parameters that determine the relative importance of the different constraints we consider.

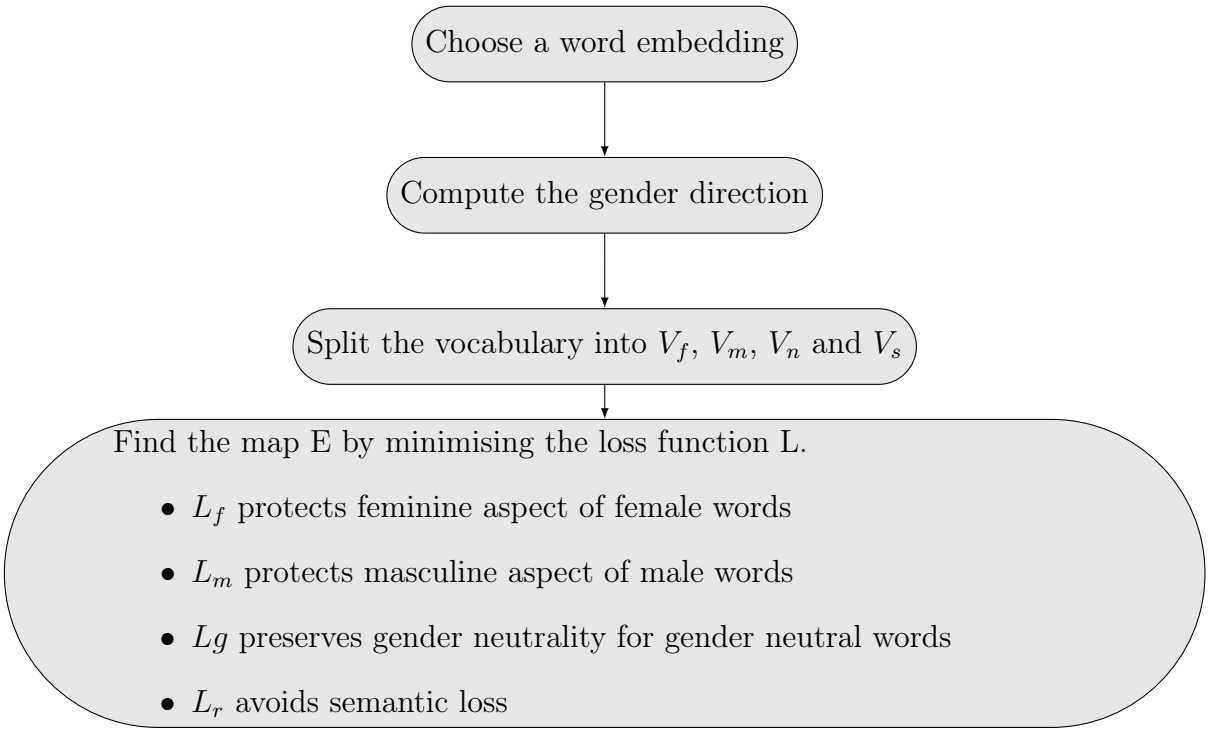


Figure 2.3: Steps of the GP-GloVe method

2.3.4 Half-Sibling Regression (HSR)

Yang et al. [41] proposed a method for reducing gender bias that uses the statistical dependency between gender-definition word embeddings and gender-biased word embeddings. Key idea of this approach is to learn and then directly subtract the gender information from the non-gender-definition words. First, words in vocabulary are classified in gender-definition (*she, he*) and non-gender-definition (*nurse, colonel*) words. Both sets contain underlying gender information, but semantic content is mostly present in non-gender-definition words. Debiased non-gender-definition word vectors are obtained as

$$\hat{V}_N \coloneqq V_N - \hat{G} \tag{2.17}$$

where V_N is the original embedding and $\hat{G}$ is the approximated gender information, estimated by

$$\hat{G} \coloneqq E[V_N|V_D] \tag{2.18}$$

V_D is the set of gender-definition word vectors. Given that V_D and V_N contain the same information about gender but V_D has no semantic meaning, only spurious gender information is learned by $\hat{G}$. Technically, $\hat{G}$ is calculated through ridge regression. Advantages of this method are that it is simple, it has only one hyper-parameter for the ride regression.

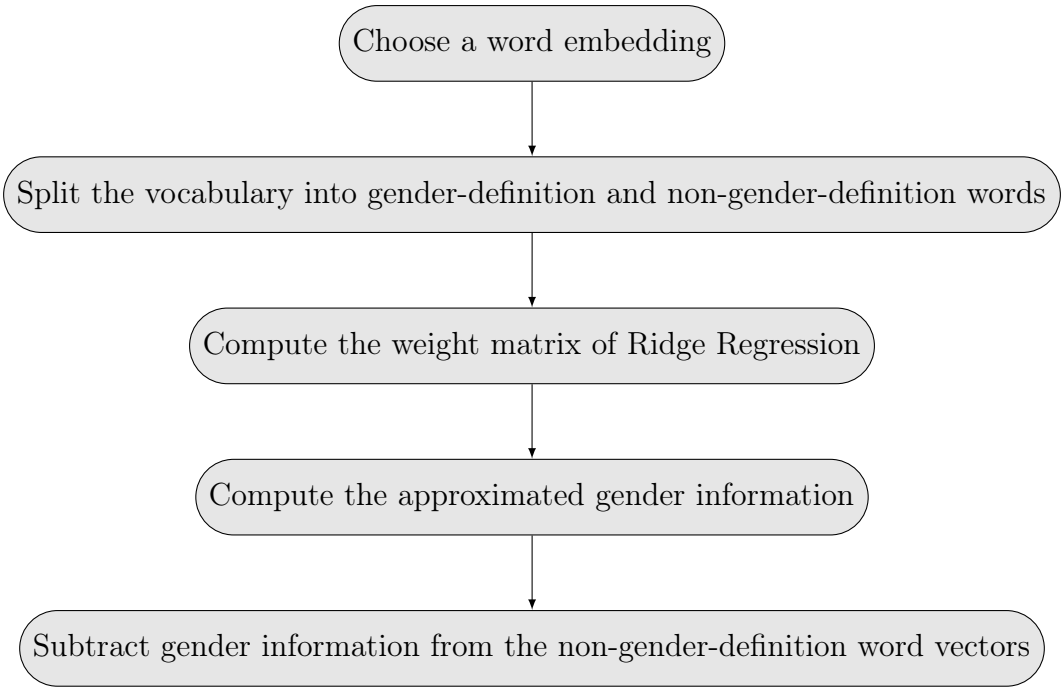


Figure 2.4: Steps of the HSR method

2.3.5 Conceptor Debiasing of Word Representation

Presented by Karve et al. [23], the *conceptor debiasing* has the peculiarity that it menages to remove multiple biases (*gender*, *racial*, ...) simultaneously. It first identifies a biased subspace through a conceptor matrix and then it applies its negation ($\neg$) to a given word to shrink its bias. It can be applied to both traditional and contextualized word embeddings.
Once defined a set of target words Z (i.e. words associated with a demographic group), the conceptor matrix is the one that minimises the followinf equation

$$\|Z - CZ\|^2 + \alpha^{-2}\|C\|_F^2 \tag{2.19}$$

where α is a hyperparameter and Z representes the embeddings of the target words. Details for the conceptor matrix and its derivation can be found in [29] [16] [28] [20]. Conceptually, C is a *soft projection matrix on the linear subspace where the word embeddings of the target words have the highest variance*. It represents the subspace of maximum bias. Once learned C, it ca be 'negated' and then applied to any word embeddings to shrink the bias direction. The negation of C can be obtained as

$$\neg C = I - C \tag{2.20}$$

It is possible to specify K different word lists that reflect different biases and then derive K *debiasing conceptors*. In the case of multiple biases, the debiasing conceptor is defined as

$$NOT(C_1 \vee \dots C_K) \tag{2.21}$$

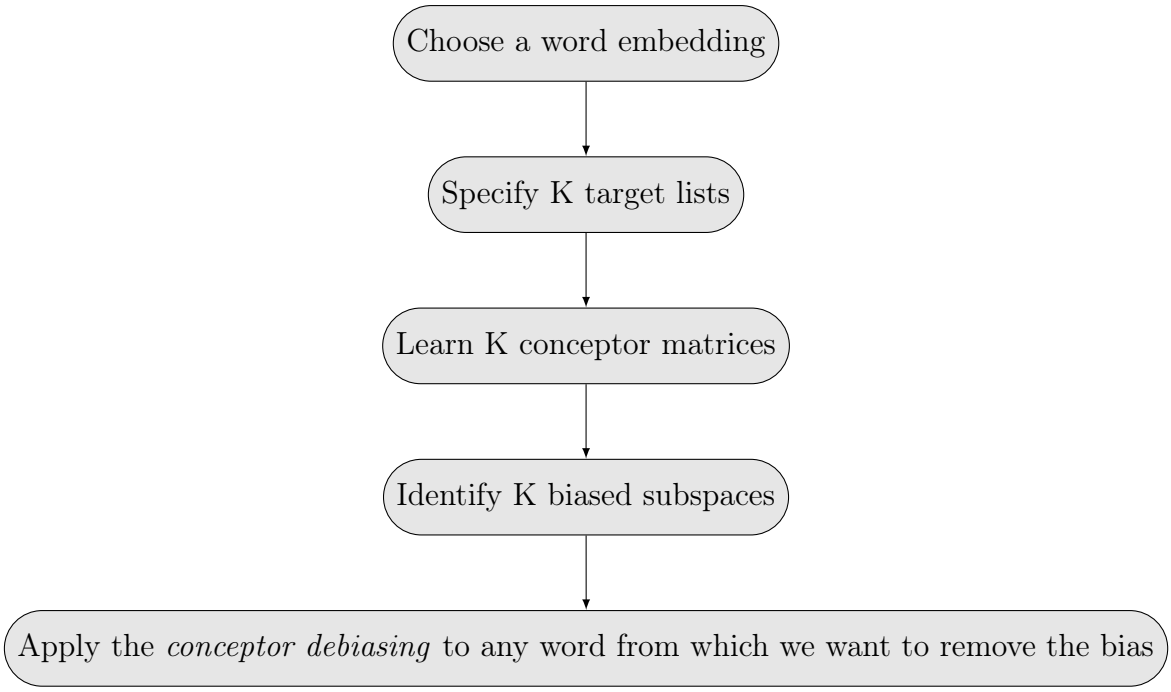


Figure 2.5: Steps of the Conceptor Debiasing of Word Representation method

2.3.6 Double-Hard Debias

Wang et al. [40] argued that word frequency in the training corpora can twist the gender direction and so it may act as harmful noise while debiasing techniques based on the identification of a gender subspace. They proposed an improvement of Hard debias method by Bolukbasi et al. by first removing the influence of word frequency and then removing the gender bias. To remove the frequency component, the 500 top male- and female-biased words are considered according to the original word embedding and PCA is performed. The top principal components are then taken and word vectors are projected into the space orthogonal to each principal component. In this intermediate subspace, the standard Hard Debias method is applied.

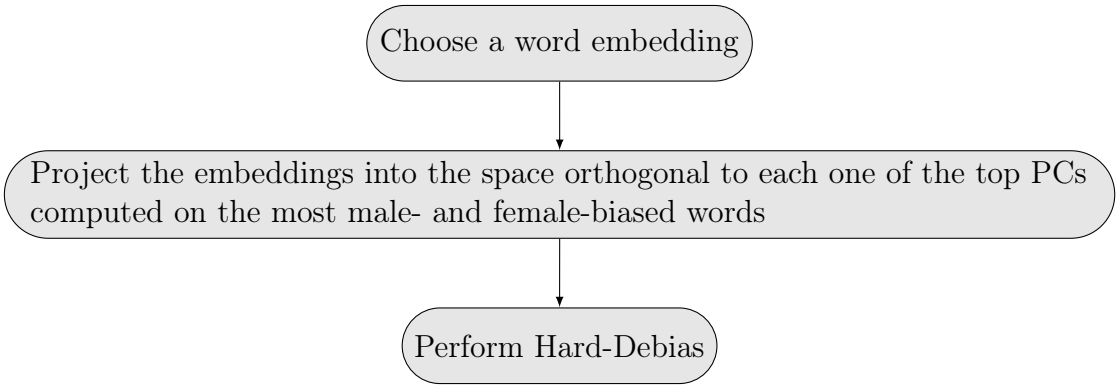


Figure 2.6: Steps of the Double-Hard-Debias method

2.3.7 Repulsion-Attraction-Neutralisation (RAN)

Another method was proposed in [27]. As always, the aim of the process is to create a transformation of a word vector that minimise the stereotypical gender information while maintaining semantic aspects. All vocabulary V is split into words for which gender carries semantic importance (i.e. *beard* and *bikini*) and all other words, which should be gender-neutral: V_p and V_d respectively.
To compute the two set from a given dictionary, authors proposed the Knowledge Based Classifier (KBC), a classifier that is based on knowledge bases instead of words embeddings. In this way they avoided adding an additional source of error as it was in the Hard-Debias method. The KCB algorithm takes as input a dictionary V, a set of dictionaries D, a set of gender-specific names *names*, a set of stop words *stw*, a set of gender-specific words *seed* and a function that checks for non-alphatibetic words *isnonalphabetic(w)*. Each dictionary $d_i \in D$ represents a definition of the word w. It works as follows:

- **Stage 1:** All stops words and non-alphabetic words are classified in V_p
- **Stage 2:** All names and words in the *seed* set are classified in V_p
- **Stage 3:** All words are classified in V_b except for those that are classified as gender-specific by more than half of the dictionaries. A word w is defined gender-specific or not based on the existence or absence of a word belonging the *seed* set in the definition $d_i[w]$.

In addition to the two kind of bias identified in [5], a third type of bias and its measure is presented. Following the results of Gonen and Golberg [15], the *gender-based proximity bias* captures the illicit proximities between a word and its closest k neighbours due to gender biases. It measured as follows:

$$\eta_{wi} = \frac{|N^b_{wi}|}{|N_{wi}|} \tag{2.22}$$

where N_{wi} represents the top k nearest neighbour according to the maximum cosine similarity and N^b_{wi} is a subset of N_{wi} with the set of neighbours with an indirect bias above a chosen threshold θ_s.

Similarly to the Gender-Preserving method, to remove the bias from a word w a function $F(\overrightarrow{w_d})$ is searched, which computes the debiased counterpart $\overrightarrow{w_d}$ of the initial word $\overrightarrow{w}$. The resulting function is given by

$$F(\overrightarrow{w_d}) = \lambda_1 \cdot F_r(\overrightarrow{w_d}) + \lambda_2 \cdot F_a(\overrightarrow{w_d}) + \lambda_3 \cdot F_n(\overrightarrow{w_d}) \tag{2.23}$$

Minimizing the three functions defines three phases that give the name to the method:

1. Repulsion: minimizing $F_r(\overrightarrow{w_d})$ leads to separate the debiased word from the neighbours with high indirect bias
2. Attraction: minimizing $F_a(\overrightarrow{w_d})$ leads to the minimum loss of semantic properties of the debiased word

3. Neutralization: minimizing $F_n(\overrightarrow{w_d})$ leads to the minimum direct bias of the debiased word

Going further into details, to remove the gender-based proximity bias from the debiased word $\overrightarrow{w_d}$ the aim is to reduce the semantic similarity between that word and a *repulsion set* S_r. This set contains all the words between the 100 nearest neighbours with an high indirect bias. This is done by minimizing the following function

$$F_r(\overrightarrow{w_d}) = \frac{(\sum_{n_i \in S_r} |\cos(\overrightarrow{w_d}, \overrightarrow{n_i})|)}{|S_r|} \tag{2.24}$$

To minimize the semantic loss, the aim to attract $\overrightarrow{w_d}$ towards $\overrightarrow{w}$ in the embedding space. This is done by minimizing

$$F_a(\overrightarrow{w_d}) = \frac{|\cos(\overrightarrow{w_d}, \overrightarrow{w}) - 1|}{2} \tag{2.25}$$

Eventually, to minimise the direct bias, the goal is to minimize the absolute value of the dot product between the word vector and the gender direction. This is done by minimizing

$$F_n(\overrightarrow{w_d}) = |\cos(\overrightarrow{w_d}, \overrightarrow{g})| \tag{2.26}$$

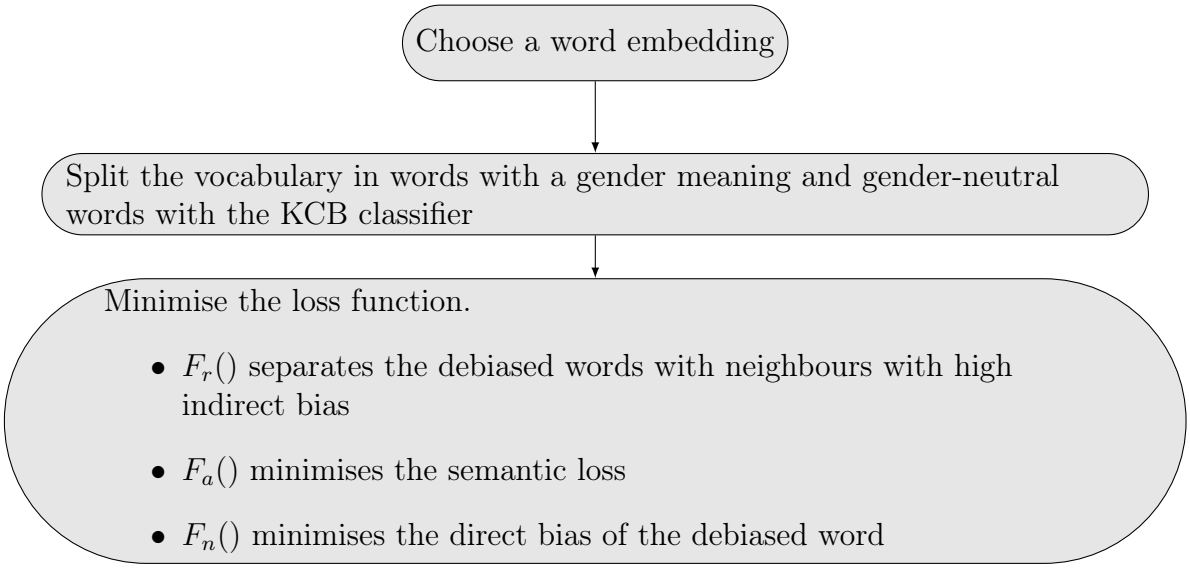


Figure 2.7: RAN steps

2.4 How to evaluate bias and word embedding

2.4.1 Word Embedding Association Test (WEAT)

The word embedding association test was proposed in 2017 by Caliskan et al. [8]. It is useful to evaluate if stereotypes are presenet in a word embedding or not. It starts from two sets of *target* words and two sets of *attribute* words. The two sets of target

words contain male and female stereotypes like *programmer, doctor, surgeon*, ... and *homemaker, nurse, beautician* ... while attribute words are gender definition words as *he, man, male, ...* and *she, woman, female, ...* The null hypothesis of the test states that the two sets of target words are related in a similar way to the sets of attribute words. If the null hypothesis is true, *homemaker* and *programmer* have approximately the same similarity with *she* and *he*, and also with all other words in the attribute words. The same holds for all target words. On the contrary, if the null hypothesis is rejected, it means that stereotypes are still relevant in the embedding. They proposed three couple of target words that define stereotypes: career and family, arts and mathematics and arts and science. In all the three experiments, two sets of attribute words, defined by a list of male name and a list of female name, are compared with the three couples of words and p-values are obtained. A word embeddings with little or no bias should not reject any of the three null hypotheses.

2.4.2 Lipstick on a Pig

In 2019, a fundamental paper in the field was written by Gonen and Goldberg [15]. The important assumption carried out by the authors is that methods for removing bias until then ([5] and [44]) are just hiding the bias and not really removing it. Gender bias is still reflected in gender neutral words and their associations. Both methods, using different approaches, remove the bias by reducing the projection of words on a gender direction. However, this does not influence similarity between gender neutral words. Gender neutral words that have a stereotypical male/female connotation, after debiasing will maintain an association due to gender, even if they are no more directly associated with it. To measure this remaining bias, they propose 5 different tasks.

Clustering task

The 1000 most biased words according to the original word embedding are selected and then clustered into two clusters with k-means. 500 male biased and 500 female biased words are considered. Each selected word has an associated *true labels* y_i that define its gender bias: 1 if the word is female biased and 0 otherwise. After performing 2-means clusters, a predicted labels $\hat{y_i}$ is assigned to each word. Accuracy measure the similarity between the computed clusters and the gender clusters. The more this clustering align with gender, the more bias is still present. Accuracy is given by

$$\frac{1}{N}\sum_{i=0}^{n-1}\mathbf{1}(\hat{y_i} = y_i) \tag{2.27}$$

To further compare the clusters with the gender classes, in this work will be also computed the adjusted rand index (ARI). The adjusted rand index compare all possible couple of units and evaluates if they are in the same cluster or not for the two different clustering solutions. When it is equal to 1, it means that all units are classified in the same cluster from both clustering. The higher the ARI, the more the clusters align with gender.

$$ARI(C_1, C_2) = \frac{R(C_1, C_2) - ER}{1 - ER} \tag{2.28}$$

where ER is the expected value of $R(C_1^*, C_2^*)$ and C_1^* and C_2^* are two independent clusters generated randomly, whose ARI would be equal to 0. $R(C_1, C_2)$ is the Rand index defined as follows:

$$R(C_1, C_2) = \frac{1}{n(n-1)/2} \sum_{\mathbf{x},\ \mathbf{y} \in D} 1(i_1(\mathbf{x},\mathbf{y}) = i_2(\mathbf{x},\mathbf{y})) \tag{2.29}$$

where $i_j(\mathbf{x},\mathbf{y} = 1)$ if $\mathbf{x}$, $\mathbf{y}$ are in the same cluster in C_j and D is the considered data set.

Correlation task

This task makes use of a new method for measuring bias that captures also indirect bias: the percentage of male/female biased words among the 100 nearest neighbours of a given word. The correlation between this metric and the bias-by-projection proposed by Bolukbasi et al. [5] helps understanding how much bias is still present in the embedding. A fair word embedding should have low correlation. To compute the bias-by-neighbours of a target word the 100 nearest neighbours are found and, through the computation of the bias-by-projection computed on the original word embedding, the number of male and female neighbours is obtained. Now, the correlation between the two measure of bias can be calculated.

Profession task

For this task the list of profession created by Bolukbasi et al. [5], that contain a list of gender-neutral professions, is used. Two plots of the professions are generated: one with the original bias on the X and the number of male neighbours before debiasing on the Y, the other with the original bias on the X and the number of male-biased neighbours (where the considered bias in the one computed on the original embedding) after debiasing on the Y. The same correlation measure as before can be used to evaluate if the distribution of words around professions has changed after debiasing.

Association task

This task consists of three experiment from Caliskan et al. [8]. They consider association between male/female words and two different set of words that reflect gender stereotypes :

- association between male/female words and family and career words
- association between male/female words and arts and mathematics words
- association between male/female concepts and arts and science words

The null hypothesis is that there is no difference between the sets of male/female words and the two sets of other words. A good result, meaning low bias, would be having three low p-values.

Classification task

The last task consists of predicting gender for gender-neutral words. After selecting the 5000 most biased words according to the original bias, both female- and male-biased, a RBF-kernel SVM classifier is trained on 1000 of them to predict the gender. Results are then evaluated on the remaining 4000 words. A high accuracy means that the predicted gender align with the gender bias. In this case, the word embedding is not free from bias.

2.4.3 SemBias data set

It is a data set build in 2018 in by Zhao et al. [44] composed by four pairs of words: a gender-definition (*wizard - which*), a gender-biased (*chef - baker*) and two other pairs (*pen - pencil, salt-pepper*). The task is to identify the gender-definition word pair from the four pairs. The data set contains 440 instances given by the combination of 20 gender-biased and 22 gender-definition pairs. A subset of 40 instances, generate by 2 gender-definition pairs not used for training, is used for testing. The gender-definition word pair is chosen to be the one with the highest cosine similarity with *he - she*. Accuracy is given by the number of times each type of word pair is selected. When a word embedding is free from gender bias, it should obtain high accuracy for gender-definition words and low accuracy for the other two kind of pairs.

2.4.4 Word Similarity task

A good word embedding should capture similarity between words as humans do. To evaluate this property some benchmark data sets are used. They are all composed by a list of word pairs and an associated similarity measure given by humans. 8 different benchmark data sets are used, each with its own measure of similarity. The *RG-65* benchmark for example contains 65 couples with an associated similarity score ranging from 1 to 5. Similarity in word embeddings is computed with cosine similarity, and then correlation between human and word embedding scores is computed.

2.4.5 Semantic Textual Similarity (STS)

The semantic textual similarity task measures the degree of semantic similarity between two texts. Given two sentences, a continuous valued similarity score on a scale from 0 to 5 is returned: 0 indicates that the two texts are independent and 5 indicates that they are equivalent. Performance is assessed by computing the Pearson correlation between machine assigned semantic similarity scores and human judgements. The used benchmark data sets are taken from the 2012 SemEval Semantic Related task (SICK) and the SemEval STS tasks from 2012 to 2015

Chapter 3

Analysis

In this chapter, some of the methods presented in the previous chapter are considered and comparative analysis are performed. The aim of this section is to understand how differently each method works and whether some methods work better than others. Understanding the degree in which the gender is removed from GloVe word embedding is also of interest. For each method, both direct and indirect bias are computed and compared. Gender information contained in word embedding is evaluated with the SemBias data set and eventually the goodness of word embedding from the semantic point of view is evaluate through the word similarity task and the semantic textual similarity.

The following analysis are performed on the GloVe word embedding. Results are coherent with the ones already present in the other papers. Specifically, [15] computed the 5 tasks for the Hard debiased Word2Vec and for the GN-GloVe; [41] reported all the following analysis for Hard-GloVe, GN-GloVe, GP-GloVe and HSR-GloVe. Here the results are extended also to GP-GN-GloVe, DHD-GloVe, RAN-GloVe and HSR-RAN-GloVe to have a full comparison. GP-GN GloVe is obtained by applying the Gender Preserving debiasing method to the GN-GloVe as done in [40] and HSR-RAN is obtained by applying the Half Sibling Regression to the RAN-GloVe.

3.1 Direct bias

The first table shows the direct bias computed on the original GloVe embedding and on the different debaised versions. As in Bolukbasi et al. [5] the 50000 most frequent words are selected from the embedding and then words with upper-case letters, digits, punctuation or longer than 20 character are excluded. Then, gender-specific words like *he* or *she*, which have a fair gender component, are excluded as well. 47698 words are eventually considered for this analysis.

Direct bias of a word is given by the cosine similarity between its embedding vector and the gender direction. Table 3.1 reports the average direct bias, as in Equation 2.1. As expected, the HARD-GloVe gets the best result. Indeed, the method is based on projecting each biased word onto the subspace orthogonal to the gender direction.

Embeddings	Mean Bias	Mean Male Bias	Mean Female Bias
GloVe	0.0375	0.0373	0.0378
HARD-GloVe	**0.0008**	0.0008	0.0009
GN-GloVe	0.0555	0.0618	0.0368
GP-GloVe	0.0366	0.0380	0.0343
GP-GN-GloVe	0.0457	0.0431	0.0482
HSR-GloVe	0.0218	0.023	0.0198
DHD-GloVe	0.0196	0.0211	0.0175
RAN-GloVe	0.0291	0.0289	0.0294
HSR-RAN-GloVe	0.0277	0.0281	0.0271

Table 3.1: Average direct bias

It is worth mentioning that GP-GN-GloVe and GN-GloVe have a higher average bias than the original GloVe. GP-GloVe has a slightly lower bias than the original GloVe. HRS-GloVe, which does not directly minimize the projection of words onto the gender direction, anyway manages to reduce direct bias. DHD-GloVe as expected is the second method in decreasing the most the direct bias because it computes similar steps as for the HARD-GloVe. RAN-GloVe and HSR-RAN-GloVe has a lower direct bias than the original GloVe but it is still present in the embedding.

A t-test for paired samples has been used for comparing all the average biases. As result, HSR-GloVe is not statistically different from the original bias at a 5% level and DHD-Glove and HSR-RAN are not significantly different from GloVe at a 1& level of confidence. DHD-GloVe gets an average bias not significantly different from the RAN-GloVe at 5% and HSR-GloVe and HSR-RAN-GloVe have not different average direct bias at a level of 1%.

Means and standard deviations are represented in Figure 3.1. Average biases are obtained by averaging the absolute values of biases, so that positive and negative values do not offset each other. Male and female average biases are also reported in Table 3.1 and displayed in Figure 3.2. For all methods, gendered averages are significantly different from each other. GN-GloVe in particular has a male direct bias much higher than the female one.

3.2 Indirect bias

3.2.1 5 tasks

As underlined by Gonen and Goldberg [15], considering direct bias is not enough to evaluate the performance of a debiasing method. Following the 5 tasks proposed in the paper and presented in the previous chapter, measures of indirect bias are reported in Table 3.2. The row related to GloVe reports the values for the 5 tasks on the original embedding, which are the worst possible.

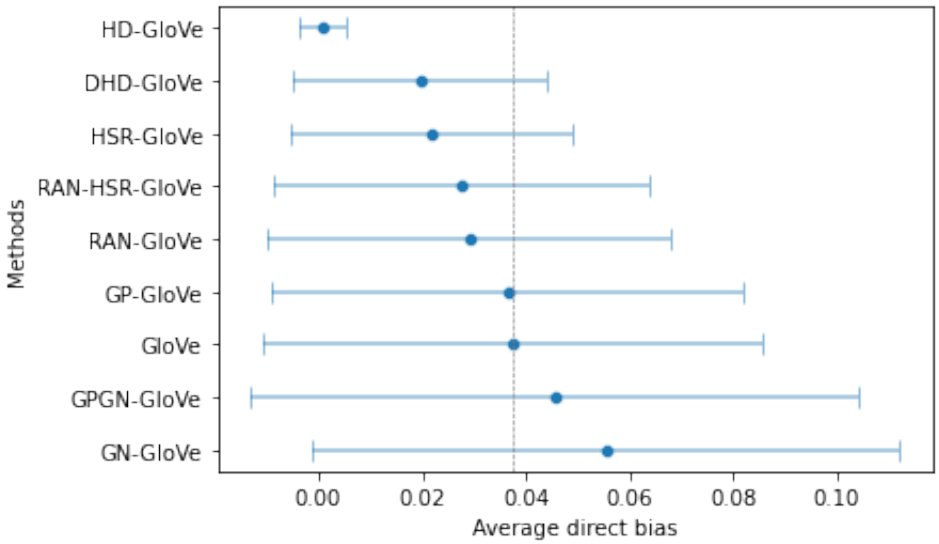


Figure 3.1: Average biases for each method with their standard deviation.

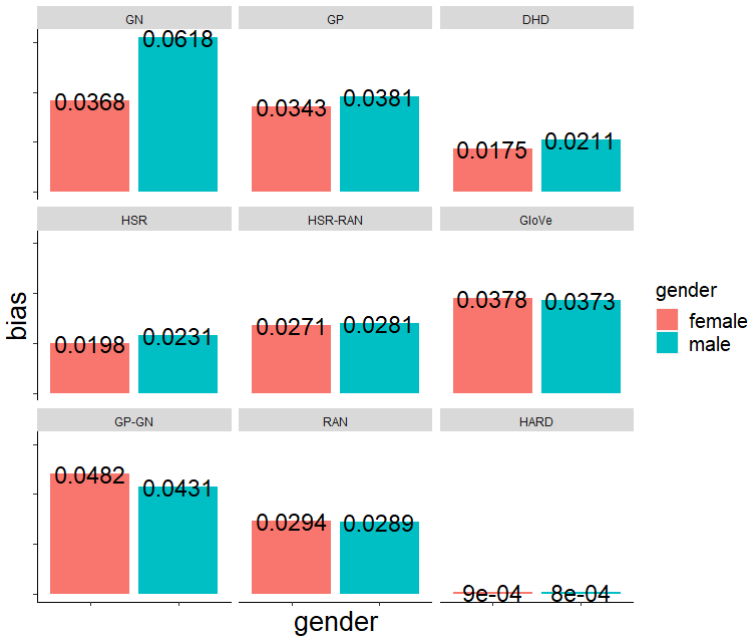


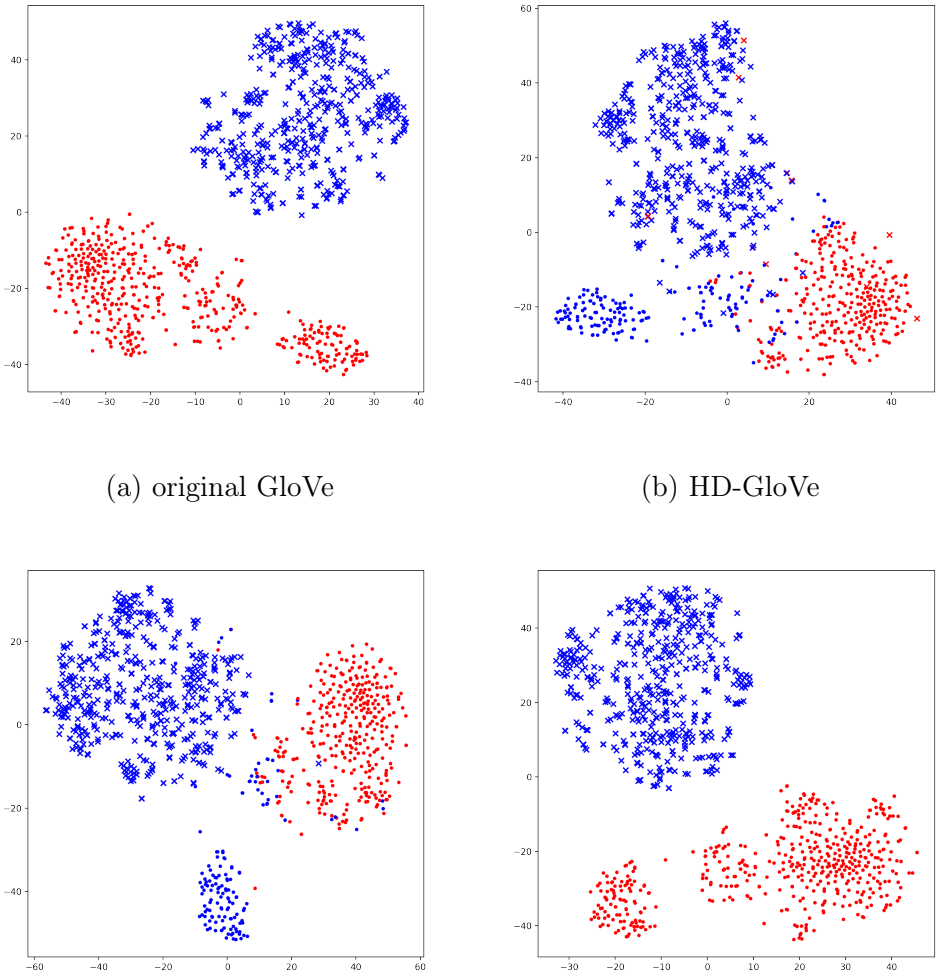
Figure 3.2: Male and Female average bias for each method.

	Clustering (Acc - ARI)	Correlation (Pear - Spea)	Profession (Pear - Spea)	Association	Classification
GloVe	1.0000 - 1.0000	0.7726 - 0.7486	0.8200 - 0.7882	2	0.9980
HARD-GloVe	0.8050 - 0.3715	0.6884 - 0.6801	0.7166 - 0.7026	**1**	0.9057
GN-GloVe	0.8560 - 0.5065	0.7336 - 0.7162	0.7925 - 0.7651	3	0.9815
GP-GloVe	1.0000 - 1.0000	0.7700 - 0.7457	0.8102 - 0.7407	3	0.9977
GP-GN-GloVe	0.8920 - 0.6142	0.7676 - 0.7408	0.8127 - 0.7840	**1**	0.9813
HSR-GloVe	0.9450 - 0.7919	**0.6422 -0.6430**	**0.6804 - 0.6733**	**1**	0.9055
DHD-GloVe	**0.7980 - 0.3546**	0.6645 - 0.6650	0.6975 - 0.6980	**1**	**0.8550**
RAN-GloVe	0.8240 - 0.4194	0.7130 - 0.6884	0.6873 - 0.6782	**1**	0.9183
HSR-RAN-GloVe	0.8170 - 0.4014	0.7043 - 0.6852	0.6983 - 0.6917	**1**	0.9383

Table 3.2: 5 Tasks results

The first column, *clustering*, reports as first value the accuracy and as second value the ARI index for all debiasing methods of the 2-means clusters solution: the lower the accuracy or the ARI, the less the clusters align with gender and the more indirect bias is removed. Plots in Figure 3.3 and 3.4 show the TSNE representation of the 1000 biased words. The original plots proposed by the authors are here improved by adding also the markers: colours are mapped to the clustering labels while markers are mapped to gender (x are female biased and . are male biased words according to the original bias). It is clear how gender and clustering labels perfectly overlaps in the original GloVe and GP-GloVe and that genders do not splits randomly in the two clusters in any method. However, the difference with respect to the clustering solution found on the original GloVe is relevant for some methods: DHD-GloVe in particular mixes cluster and gender much more with respect to other methods (Figure 3.4b). However, while the blue cluster includes mainly female biased words but also many male biased words, the red cluster contains mainly male biased words, and the same holds for HD-GloVe. In RAN-GloVe, HSR-RAN-GloVe, GP-GN-GloVe and GN-GloVe one cluster contains only male biased words. HSR-GloVe, even if it has the second higher ARI and accuracy, is the only one in which both clusters contains some female and male words, even if in each cluster one of the two gender is certainly prevalent.

Correlation refers to the correlation between direct and indirect bias. The Pearson correlation (first value) is the one considered in the original paper and the Spearman correlation (second value) is added here. All correlations are significantly different from zero. The lower correlation is found for HSR-GloVe, meaning that an originally female/male-biased word after debiasing will have not only female/male words as neighbours but also some male/female words. DHD-GloVe, RAN-GloVe, HSR-RAN-GloVe and HARD-GloVe also manage to reduce the correlation with respect to the original GloVe. GN-GloVe reduces the correlation but not much. On the contrary, GP-GloVe and GP-GN-GloVe maintain more or less the same correlation as the biased GloVe, suggesting that these methods do not remove indirect bias because words with high male/female bias before debiasing have also many male/female words around them after debiasing. Anyway, it is relevant to notice that even the HSR-GloVe, that reduces the most the correlation, still has a correlation of 64 %. This suggests that none of the proposed methods really menages to remove the gender bias from the neighbours of a given words. Even if the target word has a null direct bias, bias will still be present in its neighbour words.



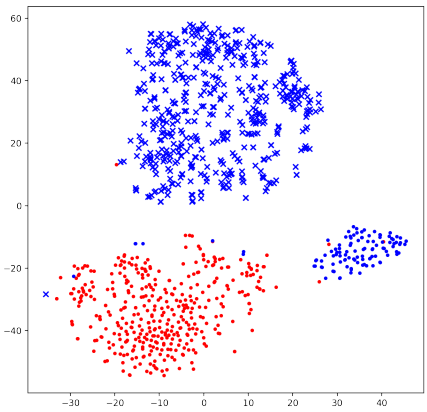
(a) original GloVe

(b) HD-GloVe

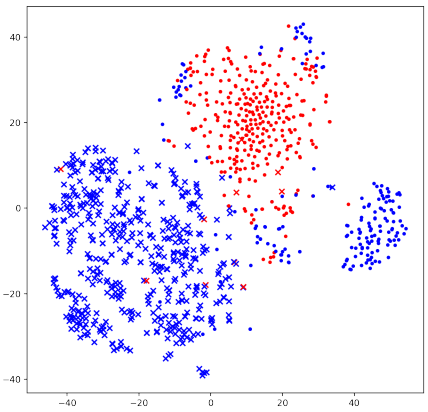
(c) GN-GloVe

(d) GP-GloVe

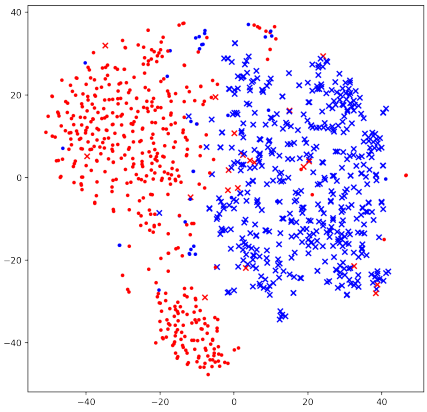
Figure 3.3: Clustering of GloVe, HD-GloVe, GN-GloVe and GP-GloVe



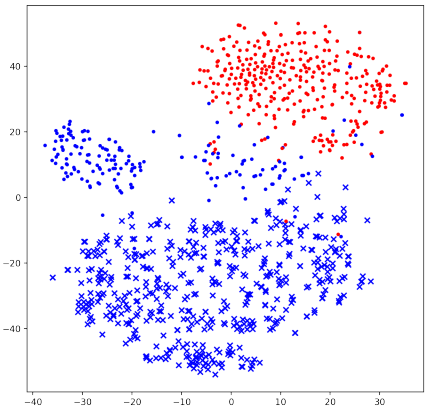
(a) GP-GN-GloVe



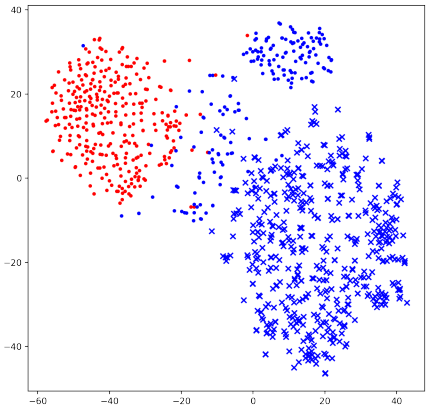
(b) DHD-GloVe



(c) HSR-GloVe



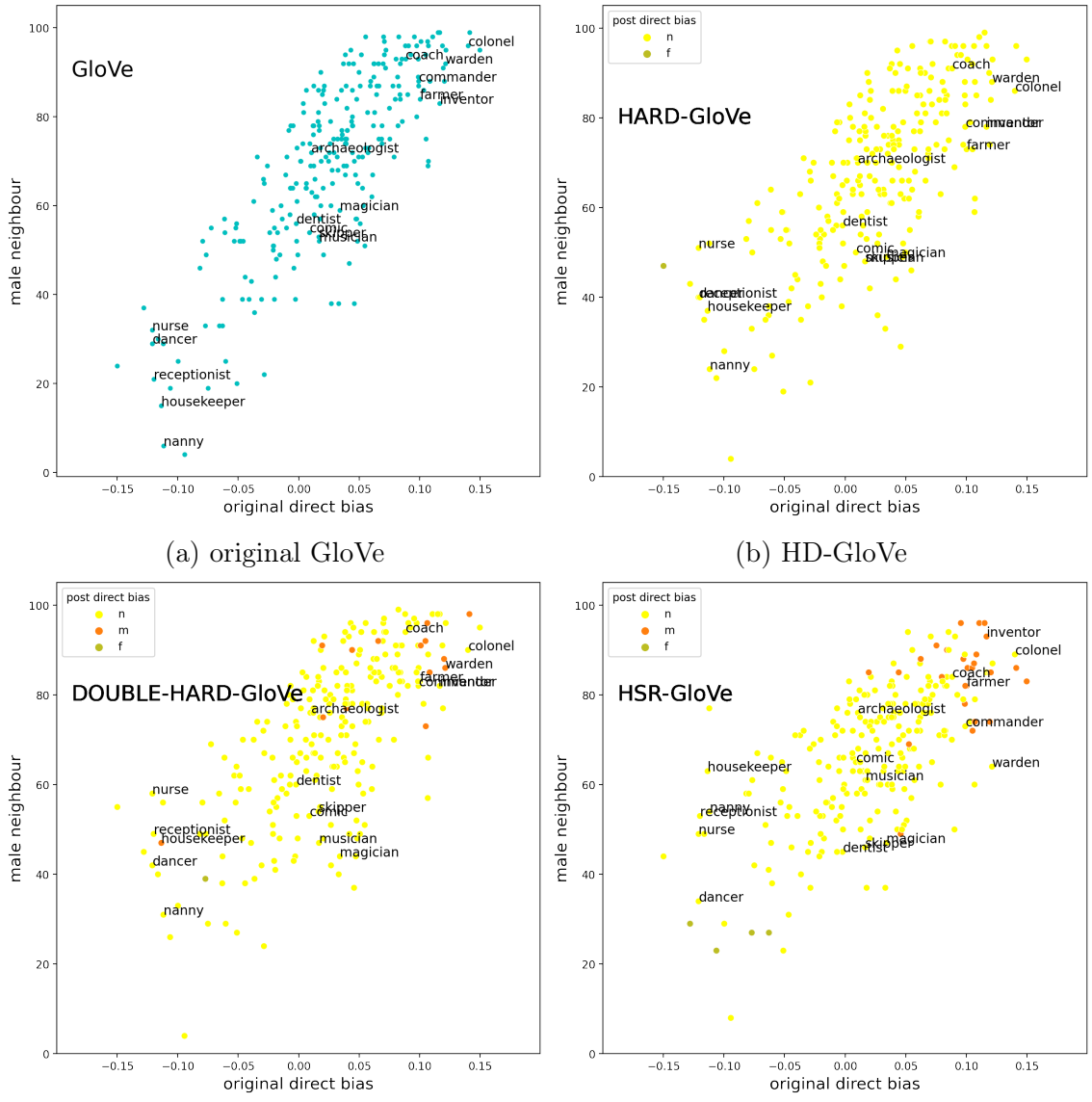
(d) RAN-GloVe



(e) HSR-RAN-GloVe

Figure 3.4: Clustering of GP-GN-GloVe, DHD-GloVe, HSR-GloVe, RAN-GloVe, HSR-RAN-GloVe

Profession contains the correlation measure between direct and indirect bias but considering only the profession words proposed by Bolukbasi et al. [5]. The best value again is for the HSR-GloVe. Results are similar as the previous task but correlations are all a little bit higher, suggesting that for the profession words removing bias from the neighbours word is harder. This is probably due to the fact that professions are one of the most stereotypical aspect of society. Graphical results are shown in Figure 3.5 and 3.6. Results on the original GloVe are shown in Figure 3.5a, while the other subplots report results on the different debised version.



(a) original GloVe

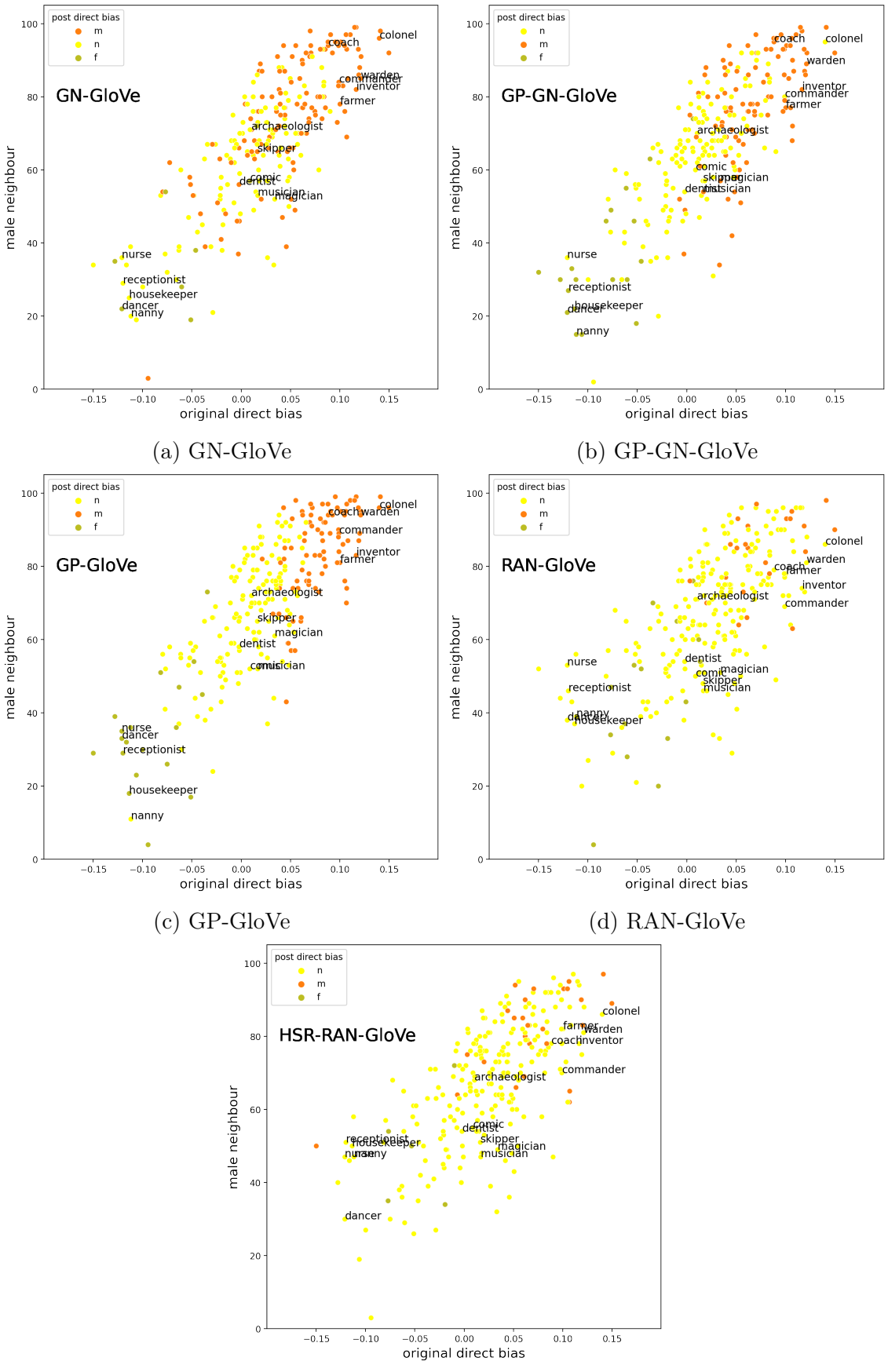
(b) HD-GloVe

(c) DHD-GloVe

(d) HSR-GloVe

Figure 3.5: Profession list of GloVe, HD-GloVe, DHD-GloVe, HSR-GloVe

It is clear that the two measures of bias are correlated for all methods. Profession with a relevant positive original bias also have many male neighbours and professions with negative bias are surrounded by female neighbours, before and after debiasing. Plots of debiasing methods show that the number of male neighbours increases for many female biased words but male biased words are still surrounded by



(a) GN-GloVe

(b) GP-GN-GloVe

(c) GP-GloVe

(d) RAN-GloVe

(e) HSR-RAN-GloVe

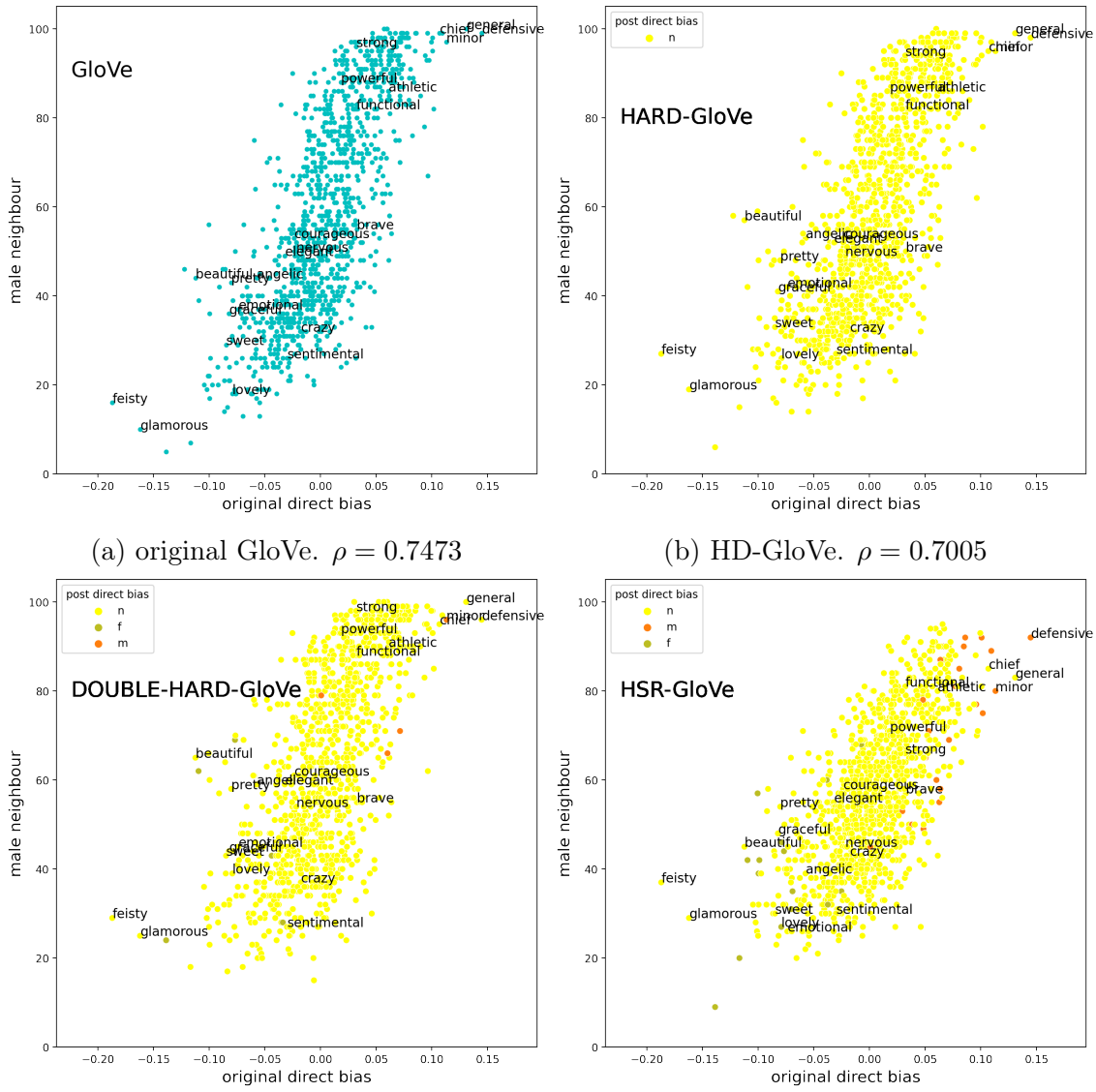
Figure 3.6: Profession list of GN-GloVe, GP-GN-GloVe, GP-GloVe, RAN-GloVe and HSR-RAN-GloVe

many more male neighbours than female. There is still a separation between female- and male- biased professions. Differently from the original paper [15], the colour assigned to the point reflects the direct bias of the debiased word embedding: yellow indicates words with a direct bias between -0.05 and 0.05, ocher words with a bias lower than -0.05 and orange stands for words with a bias greater than 0.05. Briefly, ocher shows female-biased, yellow neutral and orange male-biased words according to the new embedding. It is evident that a methods like the HD-GloVe reduces a lot direct bias but still maintain a separation between male and female stereotypical professions. The other methods, that reduce less the direct bias, show a clear correlation also between the direct bias computed on the new embedding and the number of male/female neighbours. As example, looking at results for HSR-GloVe (Figure 3.5d) it is clear that red points (male-biased words) are all in the top right of the plot while the ocher points (female-biased words) are all in the left-bottom part. The new embedding still contains neutral-words that reflects both direct and indirect bias.

To further investigate this aspect, here other lists are considered. In particular a list of adjectives and a list of sports have been created. They both contain words that should not have any gender components. The same plot done for the profession list has been computed for the other two lists: results on adjectives are reported in Figure 3.7 and 3.8 and results on the sport list are shown in Figure 3.9 and 3.10. Also in this case a relevant correlation between original direct bias and male neighbours is present. Pearson correlations are computed and reported under each figure. The method that most decreases the Pearson coefficient for both the list of adjectives and sports is DHD-GloVe. A relevant thing to notice is that GP-GN-GloVe in both cases have a correlation that is higher than the one on the original GloVe. For the list of adjectives, the same holds also for GN-GloVe. In any case, considering all methods and lists, the lower correlation coefficient is 0.6763 for the adjective lists.

The *association* column reports the number of p-values greater than 0.05 for the WEAT. A statistically significant p-value means that the null hypothesis of no difference between the two sets of target and the two sets of attribute words is rejected. There is no method that menages to get no significant p-values. This means that there is no enough evidence to prove that they are completely free from gender bias. Anyway, some methods have better results than others. In particular, HARD-GloVe, GP-GN-GloVe, HSR-GloVe, DHD-GloVe, RAN-GloVe and HSR-RAN-GloVe all have just one significant p-value. GN-GloVe and GP-GloVe are for sure the ones with worst results, because the get all p-values greater than 0.05, even worst than the original GloVe embeddings, which gets 2 low p-values.

Eventually, the last column *classification* reports the accuracy of the SVM classifier. The aim of the classifier is to classify gender after being trained on the 5000 most biased words according to the original embedding. In general, results show that accuracy is quite high in all methods. This again denotes that gender information is still trapped in the debiased word embeddings. DHD-GloVe gets the best result with an accuracy of only 85.5%. HARD-GloVe, HSR-GloVe, RAN-GloVe and



(a) original GloVe. $\rho = 0.7473$

(b) HD-GloVe. $\rho = 0.7005$

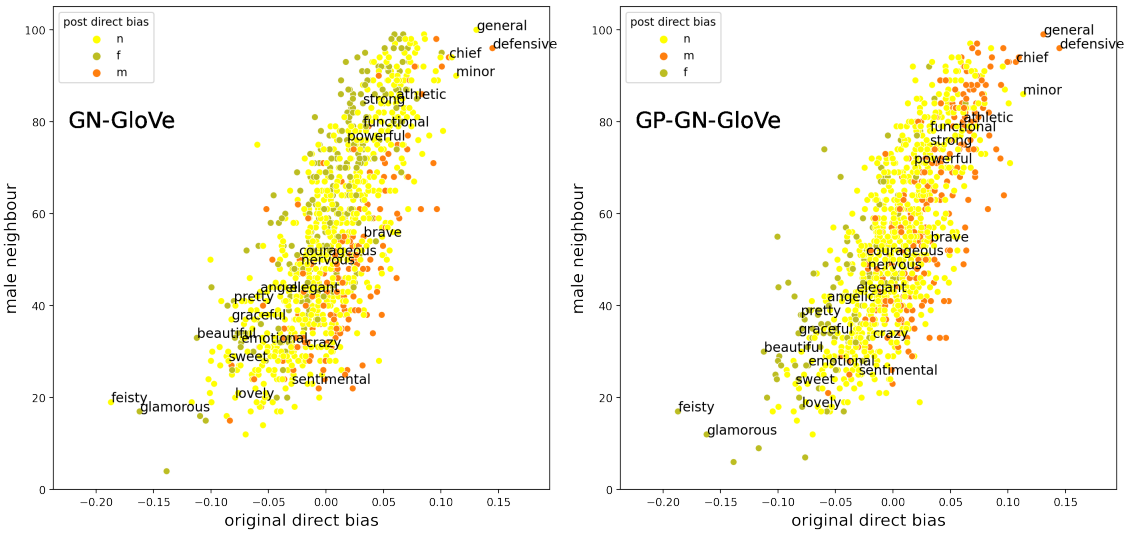
(c) DHD-GloVe. $\rho = 0.6763$

(d) HSR-GloVe. $\rho = 0.6776$

Figure 3.7: Adejective list of GloVe, HD-GloVe, DHD-GloVe, HSR-GloVe

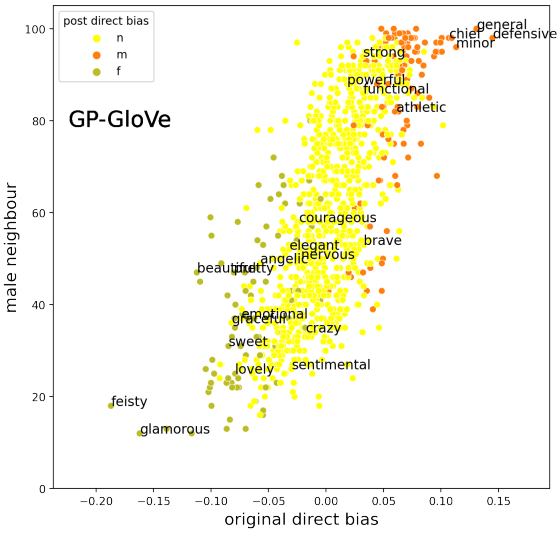
HSR-RAN-GloVe also bring to a decreasing in the accuracy with respect to the original GloVe while all other methods have a very high accuracy.

Considering all measure of indirect bias the conclusion is that there isn't a method that really outperform the others and above all there isn't a method that menage to remove completely indirect bias. DHD-GloVe, HSR-GloVe and RAN-GloVe are the one that decrease the most indirect bias anyway.

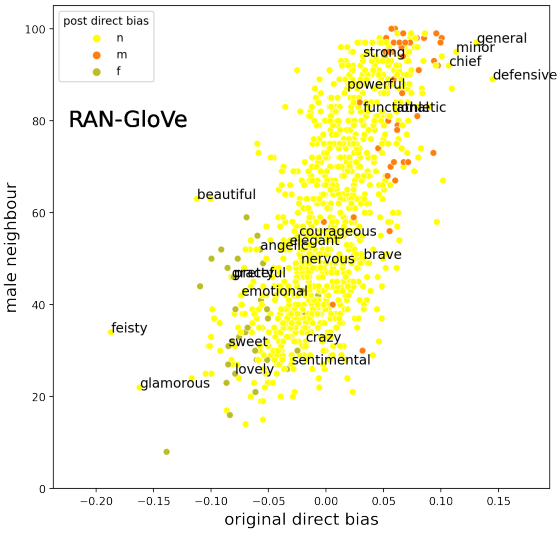


(a) GN-GloVe. $\rho = 0.8364$

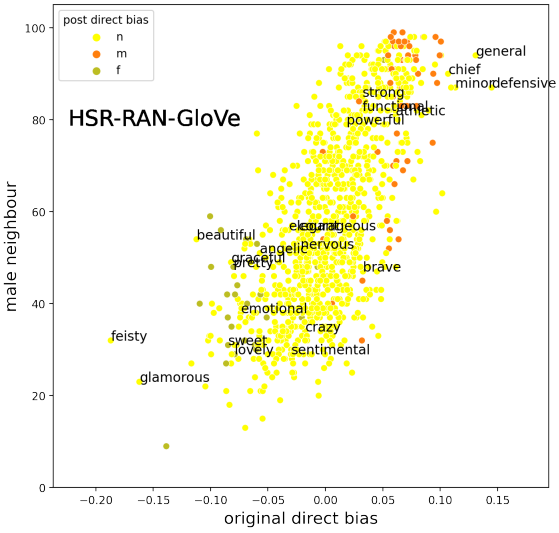
(b) GP-GN-GloVe. $\rho = 0.7667$



(c) GP-GloVe. $\rho = 0.7421$

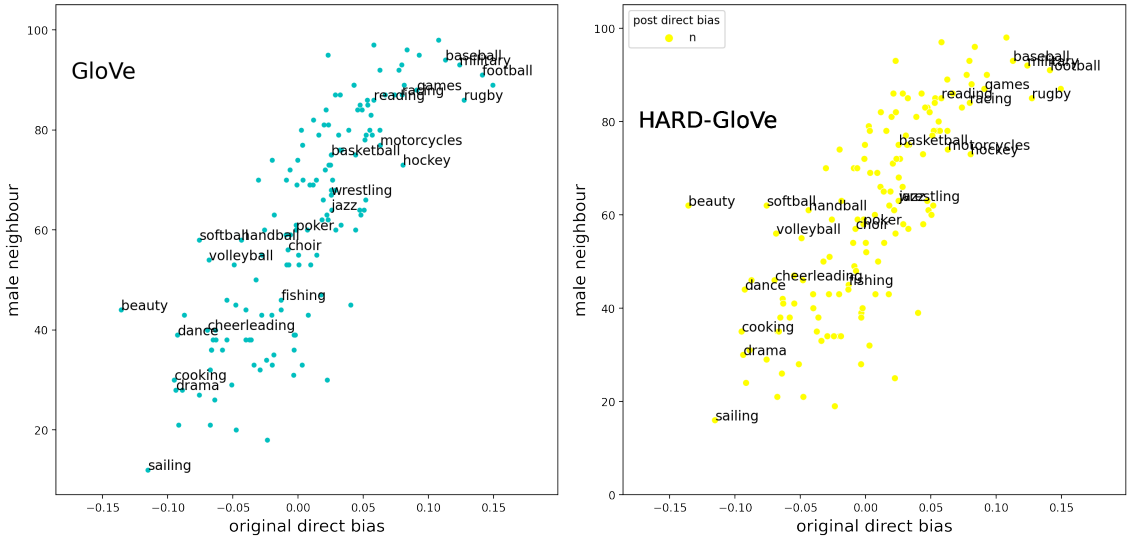


(d) RAN-GloVe. $\rho = 0.6925$



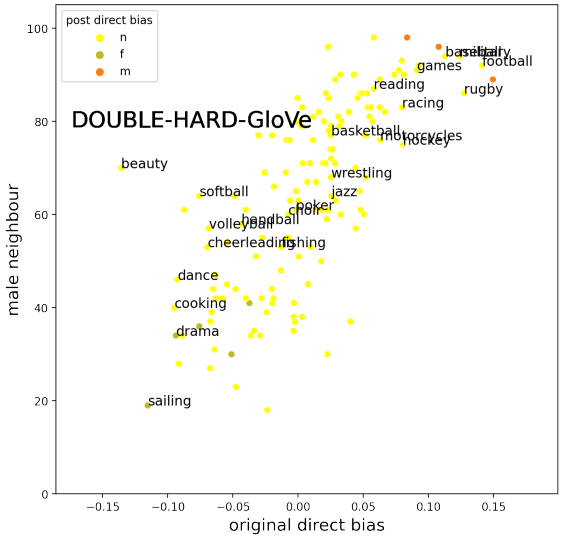
(e) HSR-RAN-GloVe. $\rho = 0.6973$

Figure 3.8: Adejective list of GN-GloVe, GP-GN-GloVe, GP-GloVe, RAN-GloVe, HSR-RAN-GloVe

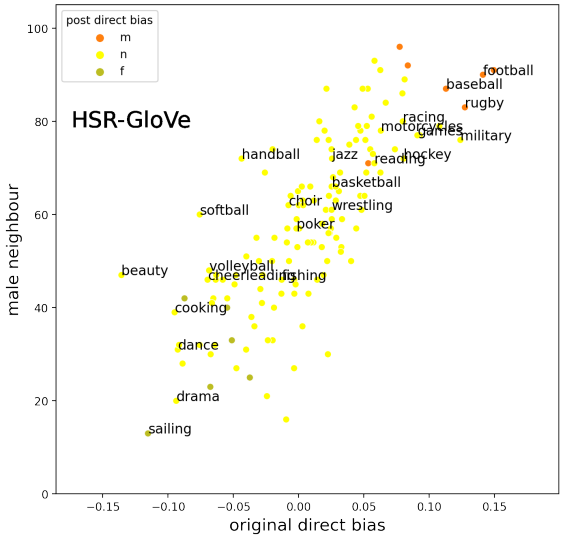


(a) original GloVe. $\rho = 0.8181$

(b) HD-GloVe. $\rho = 0.7754$

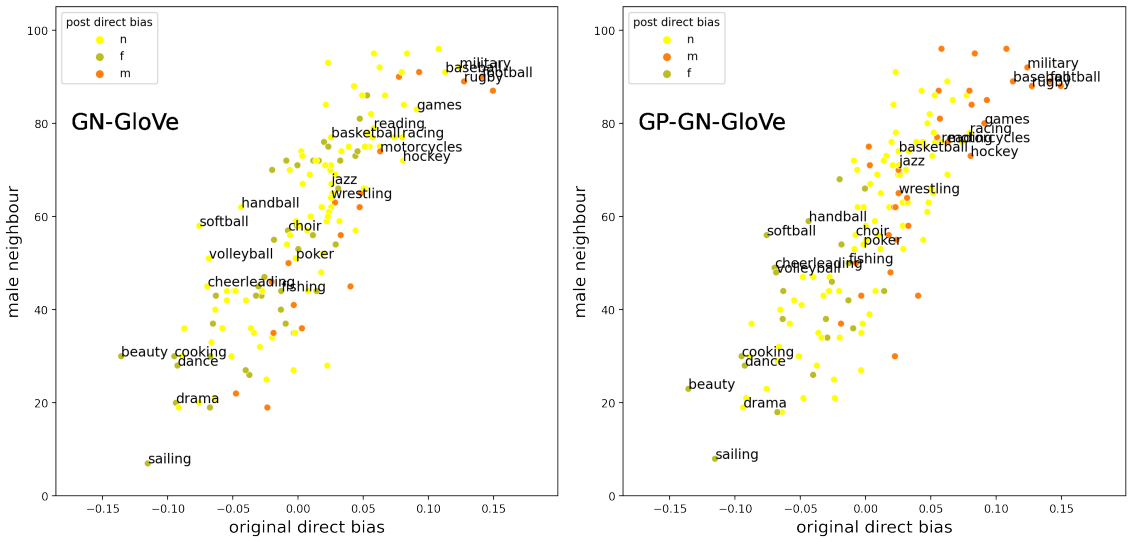


(c) DHD-GloVe. $\rho = 0.7408$



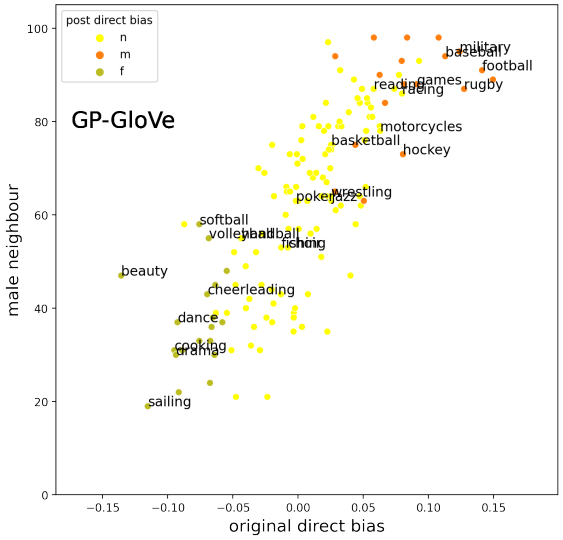
(d) HSR-GloVe. $\rho = 0.7858$

Figure 3.9: Sport list of GloVe, HD-GloVe, DHD-GloVe, HSR-GloVe

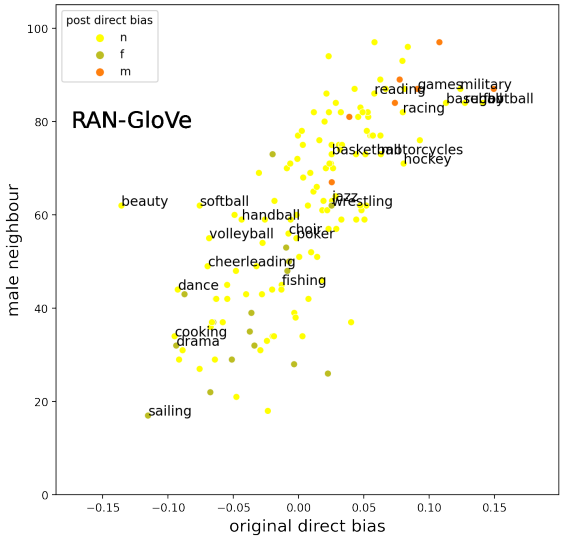


(a) GN-GloVe. $\rho = 0.7503$

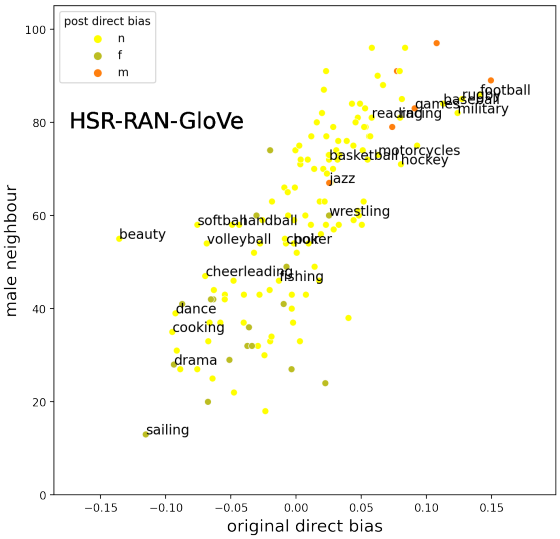
(b) GP-GN-GloVe. $\rho = 0.8416$



(c) GP-GloVe. $\rho = 0.8098$



(d) RAN-GloVe. $\rho = 0.7590$



(e) HSR-RAN-GloVe. $\rho = 0.7783$

Figure 3.10: Sport list of GN-GloVe, GP-GN-GloVe, GP-GloVe, RAN-GloVe, HSR-RAN-GloVe

3.2.2 SemBias data set

Through the SemBias data set it is possible to evaluate gender information in the word embedding. Table 3.3 reports in percentage how many times a word embedding recognizes gender-definition words and how many time it chooses gender-stereotype or other couples instead. Results under *SemBias* are the ones obtained during training and values under *SemBias subset* are results obtained on the test set. Figure 3.11 displays the same result graphically. Considering results on the test set, RAN-GloVe gets the best results with a 97.5% accuracy in identifying gender-definition pairs. It never chooses gender-stereotypes. The second best result is the HSR-RAN-GloVe, with a test accuracy of 92.5%, followed by GP-GN-GloVe, that gains the higher accuracy during training and it manages to maintain a good level also in testing and. On the other side, HSR-GloVe and DHD-GloVe do not work well at all: the accuracy for the DHD-GloVe is 0% and only 10% for HSR-GloVe for the training data. It is interesting noticing that DHD-GloVe and HSR-GloVe were the ones that most reduce indirect bias while are the worst for type of gender information.

	SemBias			SemBias subset		
Embeddings	Definition	Stereotype	None	Definition	Stereotype	None
GloVe	80.2	10.9	8.9	57.5	20	22.5
HARD-GloVe	84.1	6.4	9.5	25	27.5	47.5
GN-GloVe	97.7	1.4	0.9	75	15	10
GP-GloVe	84.3	7.9	7.7	65	15	20
GP-GN-GloVe	**98.4**	**1.1**	**0.5**	82.5	12.5	5.0
HSR-GloVe	85.9	3.8	10.2	10.0	30.0	60.0
DHD-GloVe	25.0	12.3	62.7	0.0	15.0	85.0
RAN-GloVe	92.7	1.1	6.1	**97.5**	**0.0**	**2.5**
HSR-RAN-GloVe	92.3	0.9	6.8	92.5	0.0	7.5

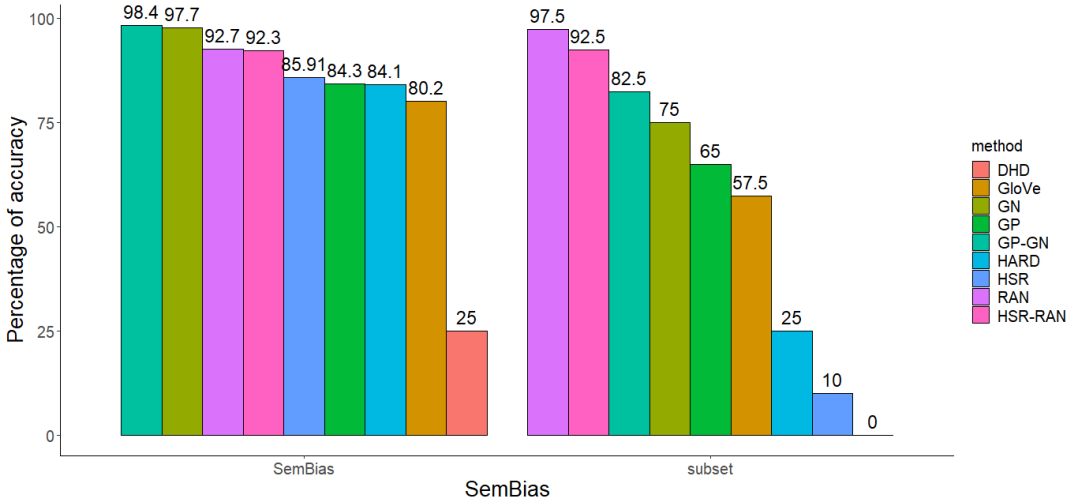
Table 3.3: SemBias results

3.3 Evaluation of word embeddings

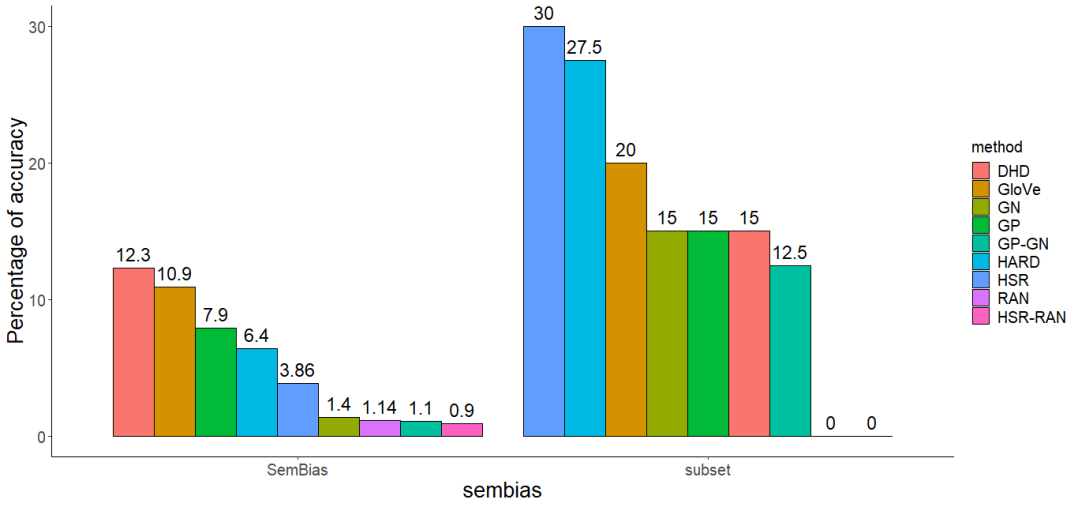
After evaluating how well a word embedding deals with gender information, it is important that it also maintains all good properties needed. As outlined in the first chapter, near words in a word embedding reflect a semantic and syntactic similarity between them. This feature should still be present after debiasing and word embedding should capture word similarities in most human-like manner.

3.3.1 Word Similarity task

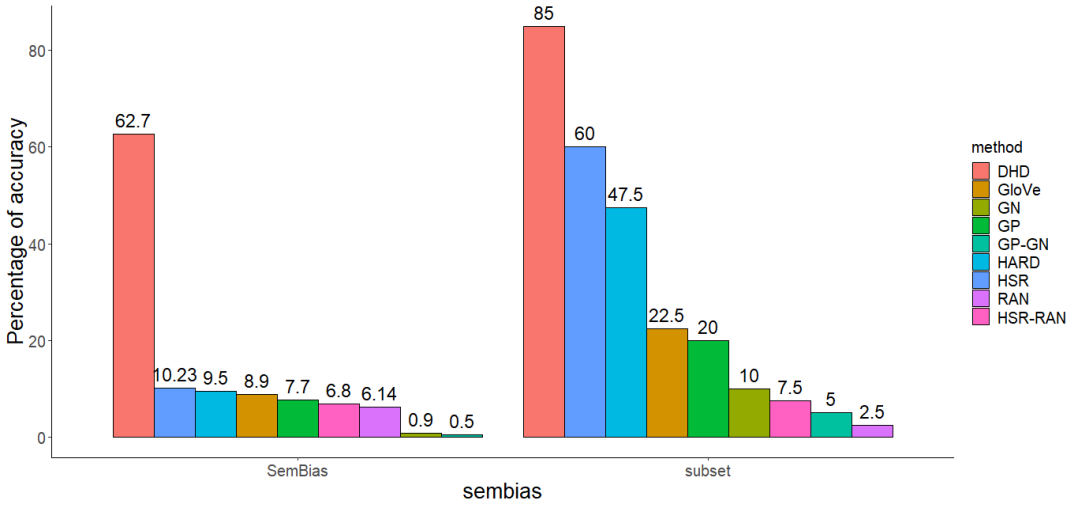
Using word similarity benchmark data sets presented in the previous chapter it is possible to evaluate if word embedding still menages to recognize similar words. Table 3.4 shows Spearman correlation coefficients between the human similarity measure given in the data sets and the similarity measure computed on the debiased word embeddings. Similarity measure is obtained through cosine similarity. The average of the correlations together with their standard deviation are reported in Figure 3.12. It is quite clear that all methods do not decrease by much the correlation



(a) Result on gender-definition word pairs



(b) Result on gender-stereotype word pairs



(c) Result on none word pair

Figure 3.11: SemBias

of the original GloVe embedding. Most of them actually menage to improve it: HSR-GloVe, GP-GN-GloVe and HSR-RAN-GloVe, which are the three methods with the highes correlations, obtain an average correlation of 0.5642, 0.5628 and 0.5615 respectively, against the average of 0.5394 of the original GloVe. Only the DHD-GloVe and the GP-GloVe have a lower average correlation than the original glove (DHD-GloVe has 0.5069 and GP-GloVe has 0.5256).

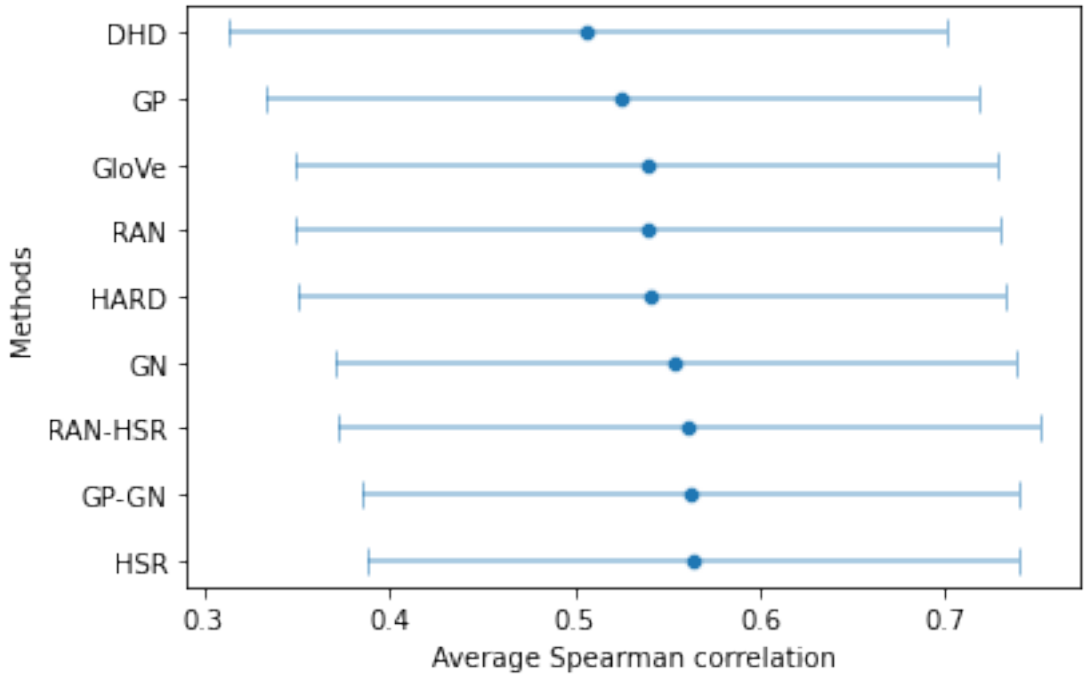


Figure 3.12: Average Spearman coefficient for WS task with standard deviation.

Embeddings	RG65	WS-353	RW	MEN	MTurk-287	MTurk-771	SimLex-999	SimVerb-3500
GloVe	0.7540	0.6199	0.3722	0.7216	0.6480	0.6486	0.3474	0.2038
HARD-GloVe	0.7648	0.6207	0.3720	0.7212	0.6468	0.6504	0.3501	0.2034
GN-GloVe	0.7457	0.6286	0.3989	0.7446	0.6617	0.6619	0.3700	0.2219
GP-GloVe	0.7546	0.6003	0.3450	0.6974	0.6418	0.6391	0.3389	0.1877
GP-GN-GloVe	0.7248	0.6355	**0.4299**	**0.7522**	**0.6650**	**0.6791**	0.3843	0.2312
HSR-GloVe	0.7764	**0.6554**	0.3868	0.7353	0.6335	0.6652	**0.3971**	**0.2635**
DHD-GloVe	0.7478	0.5699	0.3183	0.6815	0.6284	0.6175	0.3170	0.1748
RAN-GloVe	0.7651	0.6176	0.3753	0.7205	0.6462	0.6430	0.3424	0.2061
HSR-RAN-GloVe	**0.7916**	0.6445	0.3942	0.7432	0.6574	0.6630	0.3680	0.2300

Table 3.4: Word similarity task results

3.3.2 Semantic Text Similarity task

Another task to evaluate the goodness of a word embedding is the semantic text similarity task. Table 3.5 reports the average Pearson correlation coefficients between the scores given by human and word embedding to evaluate how similar two sentences are in their meaning. The same information are also reported in Figure 3.13. For this task, 4 methods get greater average correlations than the original one and 4 methods get a lower average. HSR-GloVe (average of 0.5721) gets the best

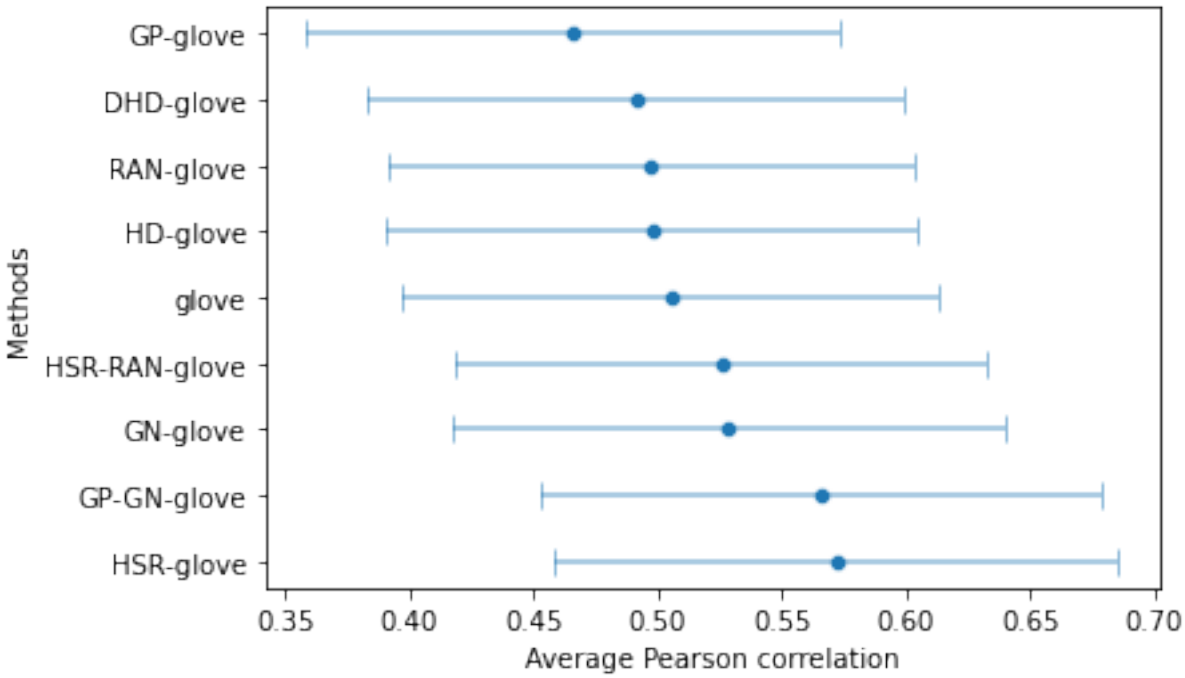


Figure 3.13: Average Pearson correlations for STS task with standard deviations

result, against the glove which has an average of 0.5051. The worst result is the one of GP-GloVe, which decreases the correlation for all the five data sets and obtains a mean correlation of 0.4659. GP-GN-GloVe performs in a similar way to HSR, improving the semantic textual similarity of the word embedding. All other methods get similar results to the original one, without bringing neither a great improvement nor a decrease.

Embeddings	STS 2012	STS 2013	STS 2014	STS 2015	SICK
GloVe	0.4892	0.4690	0.5102	0.5135	0.6211
HARD-GloVe	0.4511	0.5778	0.5838	0.4620	0.4303
GN-GloVe	0.4896	**0.6175**	0.5185	0.4869	0.5331
GP-GloVe	0.4534	0.4316	0.4670	0.4729	0.5902
GP-GN-GloVe	**0.5333**	0.5143	0.5902	0.5831	**0.6435**
HSR-GloVe	0.5127	0.5245	**0.6013**	**0.6144**	0.6256
DHD-GloVe	0.4543	0.5745	0.4766	0.4426	0.4313
RAN-GloVe	0.4806	0.4703	0.5045	0.4998	0.6090
HSR-RAN-GloVe	0.4996	0.4967	0.5387	0.5342	0.6198

Table 3.5: Semantic textual similarity task results

Conclusion

This work has focused on understanding how deeply gender bias can be removed from word embeddings, considering some of the proposed methods. Two research questions in particular have guided the analysis of this thesis.

The first question was about comparing the methods and checking if there is a significant difference among them; conclusions regarding this are not straightforward. There is no method that outperforms in all the analyses: HARD-GloVe has the lower direct bias, DHD-Glove and HSR-GloVe get the best results for an indirect bias measure, but they both perform quite bad on the SemBias dataset, where GP-GN-GloVe and RAN-Glove perform well, instead. No method leads to a decrease in performance for neither similarity or semantic tasks. Two methods have been computed by combining together two methods: GP-GN-GloVe and HSR-RAN-GloVe. In this case, results lead to similar conclusions to those of the methods used individually. However, considering all tasks, RAN-GloVe is probably the one than gets the more debiased word embedding. Especially for the SemBias data set, the RAN-GloVe word embedding gets a remarkable accuracy in identifying the gender-definition pairs. The fact that it does not confuse the gender-definition with the gender-stereotype pairs, in particular, can be very useful in real applications.

The second question was about the ability of the methods te remove the bias. Results show that none of the proposed methods managed to completely remove gender information from gender neutral words. They all perform better than the original GloVe in at least one task, but the results themselves are not satisfactory and show that information related to gender is still encapsulated in the various word embeddings. In particular, from the five tasks for detecting indirect bias, it is evident that female-biased words, considering the original bias, are more similar to each other in the new embedding that to originally male-biased words and vice-versa. This leads to the same conclusion as Gonen and Goldberg [15]: all these methods are only hiding gender bias, but not truly removing it. Gender bias is still present in how the words are represented in the embedding vectors and in their neighbours.

However, the fact that these methods are not able to completely remove the bias from word embeddings does not mean that they are useless. Almost all of them have higher score than the original GloVe in the similarity and semantic tasks. Furthermore, they reduce the original bias, both directly and indirectly. Combining one of the presented methods with a modification of the corpora and/or the machine learning algorithm may lead to a fully debiased machine learning output.

Acknowledgement

I cannot help but thank the people who have helped me in one way or another during these university years. For me, this thesis represents the conclusion of a journey. Over these years, my path has crossed with that of many, and I would especially like to thank the people who have been closest to me.

First of all, I would like to express my gratitude to Professor Elisabetta Ronchieri, who assiduously and attentively followed the writing of this thesis.

I would like to thank my roommates, official and unofficial, Tia and Bobo. I have spent some of the best evenings with you over the years. Bobo, your inexhaustible source of energy has always impressed and inspired me. Tia, I thank you for the long chats and evenings *ad impezzare* people. Promise me you will take me to *la staffa* before I move.

My stay in bologna would not have been the same without Francesca and Ilaria. I would like to thank Ilaria for her extreme availability (sorry) and for making me discover that I like Americano after all. I am grateful to Francesca for listening and advising me during our thousand aperitifs together, and also for all the Sardinian dinners of course.

I also want to sincerely thank my lifelong friends, Ludo, Ale, Race, Ira and Ele. Despite the distance, I have often felt your helpfulness and friendship. No matter we are far away, you are always part of my thoughts. A special thanks goes to Ludo, the door I will never be able to close.

A special thought can only go to my family. I must thank my sister Diletta and Filippo for the extreme generosity and affection they have always shown me. Knowing that I can always count on you helps me in every difficulty. Of course, a big thank you to my parents. Thank you for accompanying me in my studies with your advice and knowledge. You are for me a great example of intelligence and culture devoid of any arrogance.

No one has been by my side more constantly than Marcello. You don't know what joy it is to reach another milestone with you: we did it! I thank you for the precious time we have spent together, for being able to rejoice in my successes as if they were yours. You have enriched these years with so many laughs and made even the most stressful moments lighter. Knowing that from today a new journey begins with you makes this day even happier.

Bibliography

[1] Vitor Albiero, Kai Zhang, Michael C. King, and Kevin W. Bowyer. Gendered differences in face recognition accuracy explained by hairstyles, makeup, and facial morphology. 2021.

[2] Marzieh Babaeianjelodar, Stephen Lorenz, Josh Gordon, Jeanna Matthews, and Evan Freitag. Quantifying gender bias in different corpora. In *Companion Proceedings of the Web Conference 2020*. ACM, apr 2020.

[3] Finale Doshi-Velez Been Kim. Machine learning techniques for accountability. *AI Magazine*, 2021.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2016.

[5] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings, 2016.

[6] Mark Bovens. Analysing and assessing accountability: A conceptual framework. *European Law Journal*, 13(4):447–468, jul 2007.

[7] Madalina Busuioc. Accountable artificial intelligence: Holding algorithms to account. *Public Administration Review*, 81(5):825–836, nov 2020.

[8] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. 2016.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[10] Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. Queens are powerful too: Mitigating gender bias in dialogue generation, 2019.

[11] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, dec 2018.

[12] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Kate Scott, Stuart Schieber, James Waldo, David Weinberger, Adrian Weller, and Alexandra Wood. Accountability of ai under the law: The role of explanation, 2017.

[13] Joel Escudé Font and Marta R. Costa-jussá. Equalizing gender biases in neural machine translation with word embeddings techniques, 2019.

[14] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. 2017.

[15] Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them, 2019.

[16] Xu He and Herbert Jaeger. Overcoming catastrophic interference by conceptors, 2017.

[17] Kashmir Hill. Another arrest, and jail time, due to a bad facial recognition match. *The New York Times*, 2020.

[18] Dirk Hovy and Shrimai Prabhumoye. Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8), aug 2021.

[19] Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure, 2020.

[20] Herbert Jaeger. Controlling recurrent neural networks by conceptors, 2014.

[21] Khari Johnson. Dall-e 2 creates incredible imagesâand biased ones you donât see. *Wired*, 2022.

[22] Masahiro Kaneko and Danushka Bollegala. Gender-preserving debiasing for pre-trained word embeddings. 2019.

[23] Saket Karve, Lyle Ungar, and João Sedoc. Conceptor debiasing of word representations evaluated on weat, 2019.

[24] Hachim El Khiyari and Harry Wechsler. Face verification subject to varying (age, ethnicity, and gender) demographics using deep learning. *Journal of Biometrics &amp Biostatistics*, 07(04), 2016.

[25] Brendan F. Klare, Mark J. Burge, Joshua C. Klontz, Richard W. Vorder Bruegge, and Anil K. Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, dec 2012.

[26] Sangamesh Kodge and Kaushik Roy. Bermo: What can bert learn from elmo?, 2021.

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

[28] Tianlin Liu, Lyle Ungar, and João Sedoc. Unsupervised post-processing of word vectors via conceptor negation, 2018.

[29] Tianlin Liu, Lyle Ungar, and João Sedoc. Continual learning for sentence representations using conceptors, 2019.

[30] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing, 2018.

[31] 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, 2019.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[33] 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)*. Association for Computational Linguistics, 2014.

[34] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[35] Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. Reducing gender bias in word-level language models with a gender-equalizing loss function, 2019.

[36] Danielle Saunders and Bill Byrne. Reducing gender bias in neural machine translation as a domain adaptation problem, 2020.

[37] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. Mitigating gender bias in natural language processing: Literature review, 2019.

[38] Philipp Terhörst, Jan Niklas Kolf, Marco Huber, Florian Kirchbuchner, Naser Damer, Aythami Morales, Julian Fierrez, and Arjan Kuijper. A comprehensive study on face recognition biases beyond demographics, 2021.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[40] 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, 2020.

[41] Zekun Yang and Juan Feng. A causal inference method for reducing gender bias in word embedding relations, 2019.

[42] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning, 2018.

[43] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods, 2018.

[44] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. Learning gender-neutral word embeddings, 2018.

[45] Pei Zhou, Weijia Shi, Jieyu Zhao, Kuan-Hao Huang, Muhao Chen, Ryan Cotterell, and Kai-Wei Chang. Examining gender bias in languages with grammatical gender, 2019.

[46] Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology, 2019.