2022

# Measuring and Comparing Social Bias in Static and Contextual Word Embeddings

Alan Cueva Mora
*Technological University Dublin*

## Recommended Citation

Mora, A. C. (2022). Measuring and Comparing Social Bias in Static and Contextual Word Embeddings. Technological University Dublin. DOI: 10.21427/9FEA-6F46

# Measuring and Comparing Social Bias in Static and Contextual Word Embeddings

OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

**Alan Cueva Mora**

*D20125565*

A dissertation submitted in partial fulfilment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computer Science (Data Science)

**2022**

# DECLARATION

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Science), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

**Signed:** AlanCM

**Date:** **05 January 2022**

# ABSTRACT

Word embeddings have been considered one of the biggest breakthroughs of deep learning for natural language processing. They are learned numerical vector representations of words where similar words have similar representations. Contextual word embeddings are the promising second-generation of word embeddings assigning a representation to a word based on its context. This can result in different representations for the same word depending on the context (e.g. river bank and commercial bank). There is evidence of social bias (human-like implicit biases based on gender, race, and other social constructs) in word embeddings. While detecting bias in static (classical or non-contextual) word embeddings is a well-researched topic, there has been limited work in detecting bias in contextual word embeddings, mostly focussed on using the Word Embedding Association Test (WEAT). This paper explores measuring social bias (gender, ethnicity, and religion) in contextual word embeddings using a number of fairness metrics, including the Relative Norm Distance (RND), the Relative Negative Sentiment Bias (RNSB) and the already mentioned WEAT. It extends the Word Embeddings Fairness Evaluation (WEFE) framework to facilitate measuring social biases in contextual embeddings and compares these with biases in static word embeddings. The results show when ranking performance over a number of fairness metrics that contextual word embedding pre-trained models BERT and RoBERTa have more social bias than static word embedding pre-trained models GloVe and Word2Vec.

**Key-words:** *Natural Language Processing; Social Bias; Word Embeddings; Contextual Word Embeddings; Sentence Embeddings; Fairness Evaluation.*

## ACKNOWLEDGEMENTS

I would like to express my sincere thanks and admiration to my supervisor, professor Sarah Jane Delany, for her constant support, guidance and patience throughout the development of this research project.

I would like to thank Dr Emma Murphy and Dr Luca Longo for helping me with the coordination and the design of the proposal of this research project respectively.

I would like to thank the WEFE developer team headed by Pablo Badilla and Felipe Bravo-Marque from the Department of Computer Science, Universidad de Chile, which helped me in the timely completion of this work.

And finally, thanks to my loved family, in particular my mother Rosa Isela, my nieces Emma and Frida and my girlfriend Jessica Mariela, thanks for being the rays of sunshine that guide me in the darkest times.

## TABLE OF CONTENTS

## TABLE OF FIGURES

## TABLE OF TABLES

# 1. INTRODUCTION

## 1.1 Background

Natural language processing (NLP) refers to the branch of Computer Science, Artificial Intelligence and Computational Linguistics concerned with giving computers the ability to understand language in much the same way human beings can (IBM Cloud Education, 2021).

NLP combines computational linguistics (rule-based modelling of human language) with statistical, machine learning, and deep learning models. NLP tasks have become very popular because they break down the human language in ways that help the computer make sense of what it is ingesting. Some common examples of these tasks include the following:

- Sentiment analysis
- Speech recognition
- Natural language generation
- Machine translation
- Word sense disambiguation

Having so many applications the NLP field sounds promising, but it has been proved that NLP systems capture linguistic regularities that reflect social biases; human-like implicit biases based on gender, race, religion, and other social constructs (Caliskan et al., 2017). These social biases have serious consequences in their systems.

One famous example is Amazon's automated resume screening for selecting the top job candidates that turned out to be discriminating against women in 2015 (Dastin, 2018). This NLP system used resume samples of job candidates to train recruitment models and score future candidates. In consequence, women candidates were frequently discarded by the models because of the unrepresented female candidates in the training. Amazon soon abandoned the automated recruitment tool after they had discovered the bias.

Word Embeddings (Pennington et al., 2014; Mikolov et al., 2013) are dense vector representations of words and have been considered one of the biggest breakthroughs in the NPL field. The Word Embeddings are generated using Neural Network architectures trained from very large datasets, and the result is a numerical vector that represents the meaning of the word:

$$\text{WordEmbedding}(w) = \{ v_1, v_2, \ldots, v_n \}$$

Where:

- WordEmbedding is the static word embedding model.
- w is a word string, e.g. ‘she’.
- $v_x$ is the $x^{th}$ float value in the vector (word embedding).
- n is the number of elements in the vector.

These vector representations are an ideal fit with the input requirements of all Machine Learning (ML) algorithms and being a pre-trained Neural Network architecture they can be used as input layers of other Neural Networks.

These classic Word Embeddings are also called Static Word Embeddings (SWE) because the resulted vector is always the same even when the word could have many different meanings. Unfortunately, these SWE still capture the social bias of the training language and what is worse, there is scientific evidence that SWEs increase the level of bias of the training data (Bolukbasi et al., 2016).

Context is an important part of every language, especially in the English language because words can have different meanings depending on the sentence context. For example, the word bank has two different meanings in the sentences “willows lined the bank” and “they robbed the bank”. In these different scenarios, the SWE models represent both words with the same vector as if they have the same meaning.

This is a serious research gap in the SWE field. Due to the goal of the NLP, which is to understand the meaning of language, learning the context of a word was important, so new techniques for Word Embeddings were explored resulted in the Contextual Word Embeddings (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019).

These Contextual Word Embeddings (CWE) generate the vector dynamically, so it will produce a different representation of the same word depending on its context and it gets the context from the sentence. Due to this, the CWE require a sentence instead of a word and produces Sentence Embeddings:

$$\text{ContextWordEmbedding}(s) = \{ v_1, v_2, \ldots, v_n \}$$

Where:

- ContextWordEmbedding is the context word embedding model.
- s is a sentence string, e.g. ‘she is a programmer’.
- $v_x$ is the $x^{th}$ float value in the vector (sentence embedding).
- n is the number of elements in the vector.

Unfortunately, new researches proved that contextual word embeddings models also contain bias (Basta et al., 2019; Kurita et al., 2019; Zhao et al., 2019), so all the mentioned consequences related to bias are present in these new Word Embeddings models too.

Detecting and removing bias in word embeddings are typical topics in recent NLP research, The research focuses on techniques to measure the level of bias and mitigate it.

For measuring bias, many metrics were proposed like Word Embedding Association Test (Caliskan et al., 2017), Relative Norm Distance (Garg et al., 2017), and Relative Negative Sentiment Bias (Sweeney and Najafian, 2019). These metrics are called fairness metrics and the process of measuring bias is often called fairness evaluation. Also for mitigating bias, many techniques were proposed, most of them focused on a specific fairness metric.

These proposals are recent and need much more development. For example, there is not an exhaustive comparison between measuring and removing bias in SWE versus CWE, this information could be crucial if the intention of CWE is to replace SWE.

### 1.2. Research Problem

There is still a lot of research to do in detecting and removing bias in Word Embeddings. Although the final goal is to remove bias from word embeddings, first, having precise bias detection is necessary.

There are previous studies measuring the level of bias, typically gender bias, in SWE and CWE. These evaluations are performed on pre-trained word embeddings using different sources and different techniques, most of them from scratch consuming much time to the researcher, and they only consider one type of word embeddings (SWE or CWE), so there is no research about the comparison of bias between both types of word embeddings using the same implementation and metrics.

The Word Embeddings Fairness Evaluation (WEFE) framework (Badilla et al., 2020) is an optimal tool to compare and measure bias including different metrics (Word Embedding Association Test, Relative Norm Distance, and Relative Negative Sentiment Bias), models, and kinds of bias (e.g. gender, religious and ethnicity bias). Unfortunately, the WEFE framework only works with SWE models.

After analyzing the lack of evidence of comparing the level of bias between SWE and CWE, the following research question can be asked:

*"Are the levels of gender, religious and ethnicity bias, measured with the fairness metrics Word Embedding Association Test, Relative Norm Distance, and Relative Negative Sentiment Bias, lower in Contextual Word Embeddings models than in Static Word Embeddings models?"*

## 1.3 Research Objectives

The aim of this project is to measure the fairness metrics RND, RSNB, and WEAT for gender, religious, and ethnicity bias on SWE and CWE and compare them using a statistical ranking test. It is preferable to use the same fairness evaluation framework to ensure a fair comparison.

Another objective is to perform the necessary modifications on the open-source WEFE framework to be used in this experiment. The WEFE framework already processes SWE, but some modifications are needed to process CWE. Because of this, collaboration was needed with the WEFE development team headed by Pablo Badilla and Felipe Bravo-Marque from the Department of Computer Science, Universidad de Chile.

## 1.4 Scope and Limitations

The scope of this research is artificial intelligence and natural language processing, focusing on lexical semantics and its applications benefits machine translation, text supervised learning, and information extraction.

This research assumes that a fairness evaluation can be performed on pre-trained word embeddings models and the Word Embeddings Fairness

Evaluation framework is an open-source project that can be modified in order to include new functionalities. All assumptions are based on previous studies.

The main limitation for this research is the use of pre-trained Contextual Word Embeddings models. Training these models may take longer than available and unlike Static Word Embeddings, there is only one pre-trained option per each Contextual Word Embedding implementation.

This study is delimited by the fairness metrics and fairness evaluation implemented in the WEFE framework, and only gender, religious, ethnicity bias will be measured, the available time won't allow us to consider more options.

### 1.5 Document Outline

This work is structured as follows. Section 2 shows a complete exploration and explication of the literature review. Section 3 explains the approach and methodology used in the experiment. Section 4 shows the results. Finally, section 5 presents the discussion and future work.

## 2. LITERATURE REVIEW

Word embedding is the text mining technique of establishing a relationship between words in textual data (Corpus). The pre-trained word embeddings models are unsupervised neural networks learned from document corpora to capture the semantic and syntactic information about words, being a great asset for a large variety of natural processing language tasks (Oscar Deho et al., 2018). These pre-trained models receive a text as an input and generate word embeddings vectors (see Figure 1).

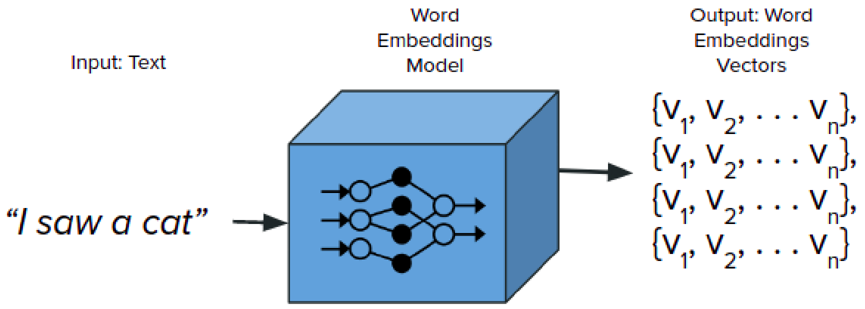


Figure 1: Generating Word Embeddings Vectors from Text. The v's are float numbers.

There are several architectures and training techniques that can be used for learning word embeddings. The great majority of them are based on the distributional semantics hypothesis: words that appear in similar contexts tend to have similar meanings. Consequently, similar words tend to be mapped to closely located vectors (Badilla et al., 2020).

The libraries produced from these different approaches are called implementations of word embeddings, and can be categorized in two: Static Word Embeddings (Pennington et al., 2014; Mikolov et al., 2013) and Contextual Word Embeddings (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019).

## 2.1. Static Word Embeddings

The static or classical word embeddings are considered static because the word embedding for a word is always the same (the numbers in the word embedding vector are the same), so words with different meanings depending on the context have the same word embedding vector. They include a vocabulary, a list of the words that can be transformed into word embedding vectors. The most used and famous implementations of Static Word Embeddings are Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014).

Word Representations in Vector Space or Word2Vec, developed by researchers at Google Inc (Mikolov et al., 2013), implements the continuous bag-of-words and skip-gram architectures for computing word embeddings vectors. The technique used to measure the quality of the resulting word embeddings in Word2Vec is the distance (similarity); words that have similar meaning tend to generate closer word embeddings vectors than those that do not.

Surprisingly, it was found that the word embedding vectors produced by Word2Vec capture many linguistic regularities, e.g. vector operations vector("Paris") - vector("France") + vector("Italy") results in a vector that is very close to vector("Rome"), and vector("king") - vector("man") + vector('woman') is close to vector("queen").

Global Vectors for Words Representation or GloVe, developed by researchers at Stanford University (Pennington et al., 2014), implements a global log bilinear regression model that combines global matrix factorization and local context window methods. This model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus, and the resulting word embedding representations showcase linear substructures of the word vector space.

As in Word2Vec, Glove measures the quality of the word embeddings using similarity metrics (Euclidean distance or cuisine similarity). This simplicity can be problematic since two given words almost always exhibit more intricate relationships than can be captured by a single number. For example, ‘man’ may be regarded as similar to ‘woman’ in that both words describe human beings; on the other hand, the two words are often considered opposites.

In order to capture in a quantitative way the nuance necessary to distinguish ‘man’ from ‘woman’, it is necessary for a model to associate more than a single number to the word pair. A natural and simple candidate for an enlarged set of discriminative numbers is the vector difference between the word pair vectors. The GloVe is designed in order that such vector differences capture as much as possible the meaning specified by the juxtaposition of two words.

The underlying concept that distinguishes ‘man’ from ‘woman’, i.e. sex or gender, should be equivalently specified by various other word pairs, such as king and queen or brother and sister. To state this observation mathematically, we might expect that the vector differences between ‘man’-’woman’, ‘king’-’queen’, and ‘brother’-’sister’ might all be roughly equal.

## 2.2. Measuring Bias in SWE

As it was mentioned in the last section, the Static Word Embeddings are based on the relationship of words in the training corpus, but this strategy has a problem, if the corpus contains social bias, it is captured by the word embeddings, and what is worse, it is increased (Bolukbasi et al., 2016; Zhao et al., 2017).

That is why different techniques and frameworks were developed to measure the bias in Static Word Embeddings. These approaches have something in common, they use a set of targets and attributes set of words to measure the level of bias.

This process is called a Fairness Evaluation and it requires a Query (targets and attributes) and a pre-trained word embedding model (Badilla et al., 2020).

### 2.2.1. Query

A query is a pair of a set of target word sets and a set of attribute word sets. All the words in a target or attribute word set should have the same concept and it is considered a term (e.g. 'she', 'woman', 'girl' are for female terms). A query sets a relationship between terms, and it is used to measure social bias, e.g. female and male terms with science and art terms is the most common query to measure gender bias. The Following is the formal definition:

$$T = \{T_1, T_2, \ldots, T_n\}$$

$$A = \{A_1, A_2, \ldots, A_m\}$$

$$Q = (T, A)$$

Where:

- Q is the query.
- $T_x$ is the x<sup>th</sup> target word set, e.g. {she, woman, girl} or {he, man, boy}.
- $A_x$ is the x<sup>th</sup> attribute word set, e.g. {math, physic, chemistry} or {poetry, dance, literature}.
- T is a set of target word sets, e.g. $\{T_{female}, T_{male}\}$.
- A is a set of attribute word sets, e.g. $\{A_{science}, A_{art}\}$.
- n is the number of sets in T.
- m is the number of sets in A.

The number of sets in T and A specifies the template of the query. Based on the last definition the template is (n, m), theoretically, n and m could have any integer value, but the next section shows that not all template queries are useful to perform a fairness evaluation. Also, a query can be split to generate subqueries with different templates, e.g. a (2,2) query can be split into two (2,1) queries and the union of these queries is the original query.

### 2.2.2. Fairness Metrics

As it was mentioned, a Fairness Evaluation requires a word embeddings pre-trained model and a query. The process of a Fairness Evaluation generates the word embeddings vectors from the word in the query sets and uses them to calculate a Fairness Metric such as the Word Embedding Association Test (Caliskan et al., 2017), the Relative Norm Distance (Garg et al., 2017) and the Relative Negative Sentiment Bias (Sweeney and Najafian, 2019).

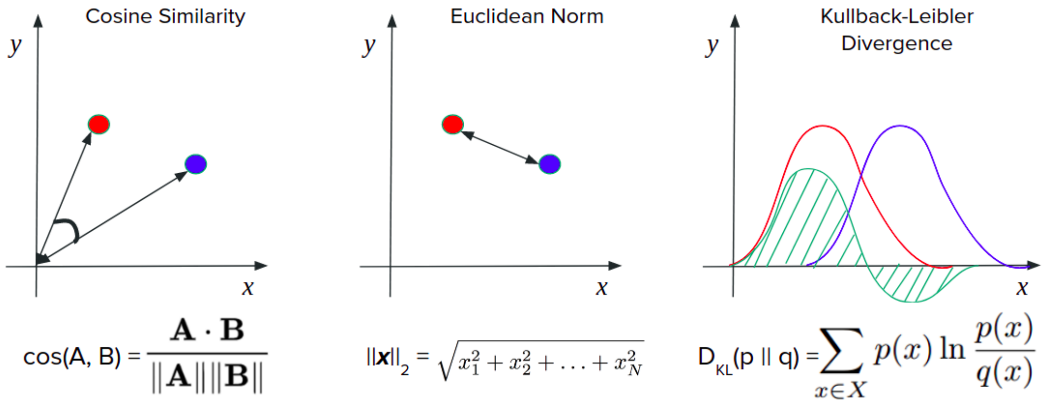

Cosine Similarity
y
x
$\cos(A, B) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$
Euclidean Norm
y
x
$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_N^2}$
Kullback-Leibler Divergence
y
x
$D_{KL}(p \| q) = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}$

Figure 2: Cosine Similarity and Euclidean Norm distance measures and the KL Divergence.

The Word Embedding Association Test or WEAT requires the word embedding vectors from a (2,2) template query (T={$T_1$, $T_2$} and A={$A_1$, $A_2$}). It is the difference of the sum of the differences of the mean of the cosine similarity of each target with respect to the attributes. The following is the formal definition:

$$F_{WEAT}(T_1, T_2, A_1, A_2) = \sum_{w \in T1} d(w, A_1, A_2) - \sum_{w \in T2} d(w, A_1, A_2)$$
$$d(w, A_1, A_2) = (\ \mathrm{mean}_{x \in A1} \cos(w, x)\ ) - (\ \mathrm{mean}_{x \in A2} \cos(w, x)\ )$$

Where:

- $F_{WEAT}$ is the WEAT fairness metric.

- $T_x$ is the xth target word embeddings vector set.
- $A_x$ is the xth attribute word embeddings vector set.
- cos(•,•) is the cosine similarity function (see Figure 2).

The idea is that the more positive the metric value, the more target $T_1$ will be related to attribute $A_1$ and target $T_2$ to attribute $A_2$. On the other hand, the more negative the value, the more target $T_1$ will be related to attribute $A_2$ and target $T_2$ to attribute $A_1$. The score that represents the absence of social bias is zero.

The Relative Norm Distance or RND requires the word embedding vectors from a (2,1) template query ($T=\{T_1, T_2\}$ and $A=\{A_1\}$). It is the sum of the difference of the Euclidean Norm between the average of the targets with respect to the attributes. The following is the formal definition:

$$F_{RND}(T_1, T_2, A_1) = \sum_{x \in A1} ( \| avg(T_1) - x \|_2 - \| avg(T_2) - x \|_2 )$$

Where:

- $F_{RND}$ is the RND fairness metric.
- $T_x$ is the xth target word embeddings vector set.
- $A_1$ is the attribute word embeddings vector set.
- $\|\bullet\|_2$ is the Euclidean Norm function (see Figure 2).
- avg(•) is the averaging of all the values in a vector.

The more positive (negative) the relative distance from the norm, the more associated are the sets of attributes towards group two (one). The score that represents the absence of social bias is zero.

The Relative Negative Sentiment Bias or RNSB requires the word embedding vectors from an (N,2) template query where N>=2 ($T=\{T_1, T_2, \ldots, TN\}$ and $A=\{A_1, A_2\}$). It creates a classifier model (logistic regression in the WEFE framework) trained from the attributes and calculates the metric from the

Kullback-Leibler Divergence of the normalized negative probability distribution of the targets (gotten from the classifier) and the uniform distribution.

$$NP = w \in T_1 \cup T_2 \ \ (C_{(A1,\, A2)}(w))$$
$$P = NP \,/\, \textstyle\sum_{x \in X} NP(x)$$
$$F_{RNSB}(P) = D_{KL}(\ P \ || \ U\ )$$

Where:

- $F_{RNSB}$ is the RNSB fairness metric.
- $T_x$ word embeddings of the target word sets.
- $A_x$ word embeddings of the attribute word sets.
- $C_{(A1,\, A2)}(\bullet)$ is a binary classifier trained with $A_1$ for negative class, and $A_2$ for positive class.
- $D_{KL}$ is an LK Divergence (see Figure 2).
- NP is the negative probability distribution of the targets.
- P is the normalized negative probability distribution of the targets, $\sum P(w) = 1$.
- U is the Uniform Distribution.

The Kullback-Leibler Divergence measures the distance over two distributions, but it is not a distance measure because it is not symmetric, so it can not be a distance metric. The Uniform Distribution is a distribution that graphically looks like a rectangle, and it is considered the expected normalized probability distribution of the targets when there is an absence of social bias, so this metric measure how far is this distribution from the uniform distribution, if they are equal (absence of social bias) the metric value is zero.

### 2.2.3. The WEFE Framework

The Word Embedding Fairness Evaluation or WEFE framework (Badilla et al., 2020) encapsulates, evaluates and compares fairness metrics. It needs a list of

Static Word Embeddings pre-trained models and a set of fairness criteria (fairness metrics), and it is based on checking correlations between fairness rankings induced by these criteria.

The WEFE framework is an open-source project and its design allows the addition of new fairness metrics, but the RND, RNSB and WEAT metrics are already implemented. Also, it includes a collection of source datasets with targets and attributes sets from previous work (Caliskan et al., 2017; Garg et al., 2017; Hu & Liu, 2004; Manzini et al., 2019).

The WEFE framework uses Gensim (Rehurek & Sojka, 2011) as a source of pre-trained models, so all the available pre-trained models in Gensim can be used in the WEFE framework.

The experiment of Badilla et al. (2020) used the WEFE framework to measure the fairness metrics WEAT, RND and RNSB for gender, ethnicity and religion bias on some different Word2Vec, GloVe, FastText, LexVec and Conceptnet pre-trained models. They use a ranking test over the metric values to compare the model's results. They conclude that the most widely used fairness metrics are not always correlated beyond the gender dimension, so more research is needed for measuring religion and ethnicity bias.

## 2.3. Contextual Word Embeddings

A research gap of the Static Word Embeddings technique is having the same word embedding representation for a word without considering the context. In English, like in every language, context is important because it can change the meaning of words, and this change of meaning could be drastic, e.g. the word “bank” in the sentence “willows lined the bank” means the land alongside or sloping down to a river or lake, while in the sentence “they robbed the bank” means a financial institution or the building of that institution.

The Contextual Word Embeddings technique solved this problem by getting a different word embedding representation of each word depending on the sentence, so the main difference between Static and Contextual Word Embeddings is that the contextual ones require a sentence to generate the word embeddings vector.

The context in the word embeddings is not exactly the same as the linguistic context. Every different sentence using the same word generates a different word embedding representation, if the linguistic context is the same those representations are similar, but never the same. That is why the Contextual Word Embeddings are also considered dynamic word embeddings.

The first Contextual Word Embeddings implementation was ELMo (Peters et al., 2018) developed by researchers at the Allen Institute for Artificial Intelligence and the University of Washington, then researchers at Google Inc developed BERT (Devlin et al., 2019), this one became very popular and some variants were developed like RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020).

Embeddings from Language Models or ELMo implements a deep bidirectional LSTM (Long short-term memory, an artificial recurrent neural network architecture) that is trained with a coupled language model objective on a large text corpus.

ELMo representations are deep, in the sense that they are a function of all of the internal layers of the bidirectional language model (biLM). More specifically, it learns a linear combination of the vectors stacked above each input word for each end task, which markedly improves performance over just using the top LSTM layer, resulting in very rich context-dependent word representations.

Bidirectional Encoder Representations from Transformers or BERT implements a multi-layer bidirectional Transformer encoder. BERT uses a masked language model procedure to train a deep bidirectional representation (left-to-right and right-to-left) by masking some percentage of the input tokens at random and then predicting those masked tokens.

BERT internally has two stages: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data over different pre-training tasks. For fine-tuning, the BERT model is first initialised with the pre-trained parameters and all of the parameters are fine-tuned using labelled data from the downstream tasks.

The popularity of BERT due to its performance in NLP tasks produces a series of variants with specific optimizations such as RoBERTa and ALBERT.

Robustly Optimised BERT Pretraining Approach or RoBERTa is a BERT variant developed to enhance the training phase, RoBERTa was developed by training the BERT model longer, on larger data of longer sequences and large mini-batches. The researchers of RoBERTa obtained substantially improved results with some modifications of BERT hyperparameters.

A lite version of BERT or ALBERT is one of the most recent BERT variants. It enhances the training and results of BERT architecture by using two techniques: Cross-Layer Parameter Sharing and Factorised embedding layer Parameterization. BERT models contain millions of parameters (about 110 million parameters in the BERT-based) which makes it hard to train, also too many parameters impact the computation. To overcome such challenges ALBERT was introduced as It has fewer parameters compared to BERT.

### 2.4. Measuring Bias in CWE

Measuring bias in Contextual Word Embeddings is more complicated than in Static Word Embeddings. Since Contextual Word Embeddings require a sentence, the proposed approaches agree neither with the word representations nor the measurement techniques. The following are some of the latest approaches to measure bias in Contextual Word Embeddings.

### 2.4.1. SEAT

May et al. (2019) proposed the Sentence Encoder Association Test or SEAT, a variant of WEAT for sentence embeddings. While the word embeddings vector is a presentation of a single word, the sentence embeddings vector is the presentation of the entire sentence. The calculation is the same as the WEAT metric but uses sentence embeddings instead, that is why it is called SEAT.

This idea of using sentences instead of words comes from the necessity of setting the correct context to the word sets, so they create the sentences replacing the words in a sentence template. Some examples of their sentence templates are “This is [WORD].”, “[WORD] is here.”, “This will [WORD].”, and “[WORD] are things.”.

These sentence templates make heavy use of deixis (general words and phrases to refer to a specific time, place, or person in context), e.g. the words “they”, “there”, “that”, “this”, etc. They are designed to convey little specific meaning beyond that of the terms inserted into them, e.g. “There is love”, “That is happy”, and “This is a friend” for the words “love”, “happy” and “friend” from the word sets of Caliskan which they used.

Their experiment measures social bias in many pre-trained models, between them ELMo and BERT, but in the particular case of measuring ethnicity bias in ELMo, some results were not statistically significant (p-value > 0.05), they interpret these results as ELMo producing substantially different representations for conceptually similar words.

The disadvantage of this approach is the addition of the sentence to set the context. The resulting sentence embeddings vector is the mean of all word embeddings in the sentence and those extra words can add noise to the embedding vector, making its comparison with word embeddings unfair, so SEAT should not be compared with WEAT.

### 2.4.2. WinoBias and 2D PCA

Zhao et al. (2019) measure and analyse gender bias in ELMo's contextualised word embeddings vectors. First, they analysed the training corpus of ELMo, the One Billion Word Benchmark (Chelba et al., 2013), and discovered that this corpus has a gender skew: male entities are nearly three times more common than female entities, which leads to gender bias.

Then, they use a sample of 400 sentences with at least one gendered word to obtain its word embeddings and apply the principal component analysis (PCA) to show that after training on such biassed corpora, there exists a low dimensional subspace that captures much of the gender information in the contextualised word embeddings (see Figure 3).

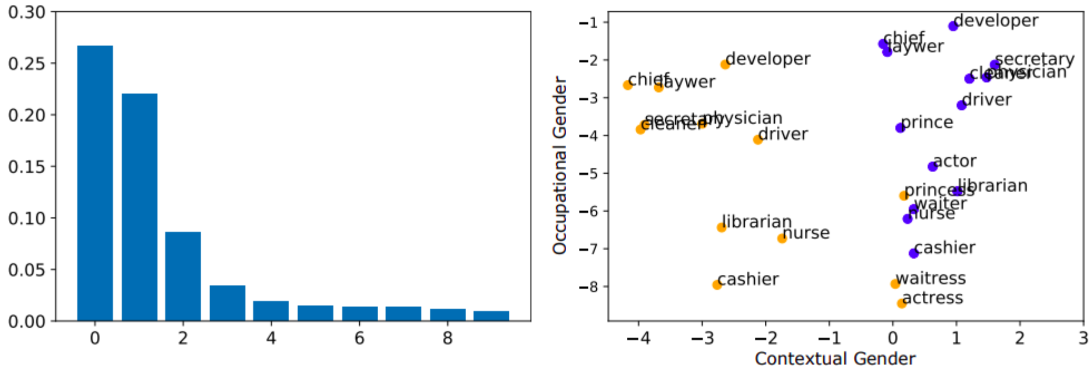


Figure 3: Left: Percentage of explained variance in PCA in the embedding differences. Right: Selected words projecting to the first two principal components where the blue dots are the sentences with male context and the orange dots are from the sentences with female context.

Thanks to figure 3 it is possible to identify two things: (1) even when the linguistic context is the same, ELMo produces different representations of words for males and females (as if they have a different context, gender context) and (2) the distance between the representations of the same word means a gender bias. It would be good to have a visualisation of the word ‘she’ and ‘he’ to identify which word is closer to what profession.

Then they measure the gender bias using a state-of-the-art coreference resolution system (Lee et al., 2018) that makes use of ELMo’s contextual embeddings on WinoBias (Zhao et al., 2018), a coreference diagnostic dataset that evaluates whether systems behave differently on decisions involving male and female entities of stereotyped or anti-stereotyped occupations.

### 2.4.3. Direct Bias

Basta et al. (2019) evaluate gender bias in Static and Contextual Word Embeddings (ELMo and Word2Vec) by calculating the fairness metric Direct Bias and a Support Vector Machine classifier model using a list of definitional pairs called ‘Definitional List’ for gender terms and ‘Professional List’ for profession terms (https://github.com/tolga-b/debiaswe/tree/master/data).

This work did not mention the concept query, but it is clear that they measure gender bias using a query of male and female terms with professional terms (the first pair are the targets and the professional terms are the attributes). This is a (2,1) template query.

In order to get word embeddings from the ELMo pre-trained model, they take representations of words by randomly sampling sentences that contain words from the Definitional List and, for each of them, they swap the definitional word with its pair-wise equivalent from the opposite gender.

Direct Bias is a fairness metric that measures how close a certain set of words are to the target vector. Similar to WEAT the distance measure is the cosine similarity, but it gets the similarity of the attributes with respect to the targets.

$$F_{DB} = 1 / |N| * \sum_{w \in N} |\cos(w, g)|$$

Where:

- $F_{DB}$ is the Direct Bias fairness metric.
- N is the number of gender-neutral words.
- g is the gender direction.
- w the word embedding vector of each word in the attribute set.
- cos(•,•) is the cosine similarity function (see Figure 2).

Their Direct Bias results show that word embeddings vectors from ELMo contain a lower level of gender bias than word embeddings vectors from Word2Vec, but for some reason, they do not include the metric values in their paper.

Then, they use a clustering approach (K-Nearest Neighbour) in 10 independent experiments to compare normal and debias word embeddings vectors. They conclude that male/female clustering, which is produced between words with strong gender bias, is less strong than in debiased and non-debiased static word embeddings.

### 2.4.4. LPBS and WEAT

Kurita et al. (2019) proposed the Log Probability Bias Score or LPBS to measure social bias in Contextual Word Embeddings pre-trained models. The LPBS fairness metric takes advantage of the masked language modelling objective of BERT models and creates simple template sentences containing the attribute (e.g. programmer) and the target words for bias (e.g. she for gender). Then mask the attribute and target tokens sequentially, to get a relative measure of bias across target classes (e.g. male and female). Contextualised word embeddings for a given token change based on its context.

For example, to compute the association between the target male gender and the attribute programmer, we feed in the masked sentence “[MASK] is a programmer” to BERT, and compute the probability assigned to the sentence “he is a programmer”. To measure the association, however, we need to measure how much more BERT prefers the male gender association with the attribute programmer, compared to the female gender. Finally, the difference between the normalised predictions for the words “he” and “she” can be used to measure the gender bias in BERT for the programmer attribute.

In order to measure the effectiveness of this new metric, they calculate the WEAT metric in BERT. For this, they use multiple sentence templates, such as “TARGET is ATTRIBUTE”, to set a specific context to the word embeddings. Table 1 shows the exhaustive list of templates used for each category.

| Category | Templates |
|---|---|
| Pleasant/Unpleasant (Insects/Flowers) | T are A, T is A |
| Pleasant/Unpleasant (EA/AA) | T are A, T is A |
| Career/Family (Male/Female) | T likes A, T like A, T is interested in A |
| Math/Art (Male/Female) | T likes A, T like A, T is interested in A |
| Science/Art (Male/Female) | T likes A, T like A, T is interested in A |

Table 1: Sentence templates used for the WEAT tests in Kurita et al. (2019) (T: target, A: attribute).

Also, they calculate the WEAT on GloVe to validate their implementation. Their results show that the level of social bias in BERT is lower than GloVe, but WEAT for BERT fails to find any statistically significant biases ($p < 0.01$) while the results of LPBS for BERT were statistically significant.

They conclude that WEAT is not an effective measure for bias in BERT embeddings, or their WEAT method requires additional investigation while their method of querying the underlying language model exposes statistically significant association across all categories, showing that BERT does indeed encode biases and that our method is more sensitive to them.

### 2.5. Reducing Bias

There are some methods to remove social bias or debias from word embeddings models. Most of those methods have been proposed with a measuring social bias approach, so these two subfields are related at the point that some gaps in the literature include the failure of these methods because they are designed for a specific measure bias approach, so the actual effect is mostly hiding the bias, not removing it (Gonen & Goldberg, 2019).

That is why even when this work focuses on measuring bias and not on removing bias, it is important to understand some of these techniques.

Data Augmentation (Zhao et al. 2018) is a method to reduce gender bias in coreference resolution by augmenting the training corpus for this task. Data augmentation is performed by replacing gender revealing entities in the training corpus with words indicating the opposite gender and then training on the union of the original data and this swapped data. In addition, they find it useful to also mitigate bias in supporting resources and therefore replace standard GloVe embeddings with bias mitigated word embeddings from Bolukbasi et al. (2016).

Neutralisation (Bolukbasi et al. 2016) is the method that instead of modifying the training corpus modify the word embeddings vectors directly by nullifying the information in the gender subspace for words that should not be associated with gender, and also equalise their distance to both elements of gender-defining word pairs. Zhao et al. (2019) apply this method on ELMo to obtain contextualised word representations of the original and the gender-swapped sentences and use their average as the final representations.

Double-Hard Debias (Wang et al., 2020) is a post-hoc gender bias mitigation technique that purifies the word embeddings against semantic-agnostic corpus regularities (e.g. word frequency) prior to inferring and removing the gender subspace. It is based on its predecessor Hard Debias (Bolukbasi et al. 2016), a method that seeks to remove the component of the embeddings corresponding to the gender direction.

Double-Hard Debias consists of two steps. First, it projects word embeddings into an intermediate subspace by subtracting component(s) related to word frequency. This mitigates the impact of frequency on the gender direction. Then it applies Hard Debias to these purified embeddings to mitigate gender bias.

Finally, Kumar et al. (2020a) proposed the Fair Embedding Engine, a library for analysing and mitigating gender bias in Static Word Embeddings. This work establishes that the focus of WEFE is limited because of its lack of support for debiasing methods, so they develop the FEE library that implements three debias methods: HardDebias (Bolukbasi et al., 2016), HSRDebias (Yang and Feng, 2020), and RANDebias (Kumar et al., 2020b). They also implement some fairness metrics such as WEAT and DirectBias.

### 2.6. Conclusion

Measuring social bias in word embeddings is a nascent topic. While measuring social bias in Static Word Embeddings is a well developed and almost standardised topic, for Contextual Word Embeddings the necessity of a method to get word embeddings vectors from pre-trained models produces a variety of approaches. It complicates the comparison between them and the approaches for Static Word Embeddings.

Also, the comparison of social bias between Contextual and Static Word Embeddings is unclear, the results show that contextualised word embeddings have a lower level of social bias, but the p-values make these results not statistically significant. Another problem is the limited number of fairness evaluations in these experiments.

If the field of removing social bias depends on a correct measure of social bias, and it is unclear that the second generation of word embedding techniques contains a lower level of social bias, to advance in these fields first more research for measuring social bias is needed.

# 3. APPROACH AND METHODOLOGY

The aim of this work is to compare the level of social bias between Static and Contextual Word Embeddings pre-trained models using the WEFE framework (Badilla et al., 2020). For this is necessary the following steps:

1. Select the target and attribute words for the queries.
2. Download the pre-trained models.
3. Get the word embeddings from the models.
4. Calculate the fairness metrics.
5. Compare the fairness metrics.

Selecting the target and attribute word sets for the queries is a crucial step because as it was explained in section 2, they directly influence the fairness metrics results. The target and attribute word sets for this experiment were collected from different sources (see section 3.1.2) and stored in a new dataset (see section 3.1.4). Since these queries needed to be used in Contextual Word Embeddings pre-trained models the current definition of a query (see section 2.2.1) was not enough, so a new type of query was proposed and used (see section 3.1.1).

Downloading the pre-trained models' steps depends on external resources. In the case of the Static Word Embeddings models, the WEFE framework already uses the Gensim library (Rehurek & Sojka, 2011). The Gensim library allows downloading many pre-trained models, so GloVe and Word2Vec were chosen because they are considered the most commonly used implementations.

After a search and analysis of libraries to download Contextual Word Embeddings pre-trained models, the Simple Transformers library (Rajapakse, 2020) was chosen because the library facilitates different combination strategies (more details in section 3.2). Unfortunately, the available pre-trained models in the Simple Transformers library are limited, so the BERT and RoBERTa implementation were chosen because of their popularity in the NLP task.

Getting word embeddings from pre-trained models is very straightforward for Static Word Embedding, but it is a real challenge for Contextual Word Embeddings. While for the static ones the method is already implemented in the WEFE framework, for the contextual ones it is not. The WEFE framework was extended in order to support contextual word embeddings (for more details check section 3.5). All changes to the WEFE framework were discussed with the WEFE Development team (see section 3.5.1).

The Calculating and Comparing the fairness metrics steps were based on the Badilla et al. (2020) experiment. The chosen fairness metrics were the RND, RNSB, and WEAT all available in the WEFE framework. Once the fairness metrics were calculated a rank test was performed over the results and the rankings were used to compare the models.

### 3.1. Query Dataset

As it was mentioned in section 2.2.1, a query is a pair of target and attribute word sets and a fairness evaluation is an evaluation of a query in a word embedding model that produces a fairness metric. The fairness metric is directly influenced by the query that is why it is important to specify the query used in an experiment.

This project aims to measure gender, religion and ethnicity bias in word embeddings, so multiple queries are needed. Measuring a social bias type using only one query is not recommended, normally an experiment uses a set of queries for each social bias type, e.g. Badilla et al. (2020) used 7 queries for gender, 9 queries for ethnicity and 9 queries for religion bias (these targets and attributes were collected from previous experiments).

In addition, the fairness evaluation executed in this experiment requires a string sentence template to set a context and relationship in target and attributes for each query, e.g. '[TARGET] is [ATTRIBUTE]'. Next section 3.1.1 develops into this concept.

Therefore, it was necessary to create a new query dataset based on datasets from related work with the necessary additions and preprocessing for this experiment requirements.

### 3.1.1. Contextual Query

In a fairness evaluation for Static Word Embeddings models, the targets and attributes word sets of a query can be processed (transform them to word embeddings) separately because the context is not needed. In the case of Contextual Word Embeddings models, it is common to use a sentence to specify the context of the target or attribute word.

Based on the work of Kurita et al. (2019) a template sentence string is used where two tags will be replaced by the target and the attribute, setting a specific context for both words (see Figure 4). Then the produced sentence is used to get the word embeddings (see section 3.2.2).

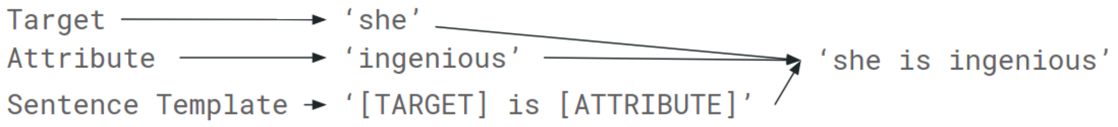


Figure 4: Sentence Template Example.

The target and attributes in a query have to be related to a topic, e.g. Male and Female terms with Science and Art terms, so it is possible to complement this relation with a sentence template, e.g. Singular Male and Female terms with Science and Art and the sentence template '[TARGET] likes [ATTRIBUTE]'.

This new concept of a query is useful to measure social bias in Contextual Word Embeddings.

Therefore, this work proposes a new query variant, a Contextual Query (CQ) where a fairness evaluation using a contextual query on a Contextual Word Embedding model should be equivalent to a fairness evaluation using a classic query with the same targets and attributes on a Static Word Embedding model. The following is the formal definition of a contextual query:

$$CQ = \{T, A, ST\}$$

Where:

- T is a set of target word sets, e.g. {{she, woman, girl}, {he, man, boy}}.
- A is a set of attribute word sets, e.g. {{math, physics, chemistry}, {poetry, dance, literature}}.
- ST is a sentence template string, e.g. [TARGET] likes [ATTRIBUTE].

### 3.1.2. Collecting Data

Once the necessary data to define a context query was specified (targets, attributes, and sentence template), the following step is to collect it to produce a number of queries. The WEFE framework offers some source datasets collected from the literature review. Table 2 shows the used source datasets and word sets used to create our query dataset and their respective literature.

| Source | Target sets | Attribute sets |
|---|---|---|
| WEAT (Caliskan et al., 2017) | male_terms, female_terms, male_terms_2, female_terms_2 | career, family, math, art, science, arts_2, pleasant_5, unpleasant_5, weapons, instruments, pleasant_9, unpleasant_9 |
| RND (Garg et al., 2018) | male_terms, female_terms, names_white, names_black, | adjectives_intelligence, adjectives_appearance, adjectives_sensitive, male_occupations, |

| | | |
|---|---|---|
| | names_asian, names_hispanic, names_chinese | female_occupations, occupations_white, occupations_black, occupations_asian, occupations_hispanic |
| Sentiments (Hu & Liu, 2004) | | positive_words, negative_words |
| Debias Multiclass (Manzini et al., 2019) | christianity_terms, islam_terms, judaism_terms | male_roles, female_roles |

Table 2: Query dataset sources.

These targets and attributes are useful to perform a fairness evaluation in Static Word Embeddings, but they need some manual changes in order to adapt those queries to the contextual query definition. Some targets were split manually, e.g. female terms in singular female and plural female terms.

Those word sets (targets and attributes) are used more than one time to create multiple queries (e.g. male and female terms with science and art and male and female terms with career and family). For more details about the final version of the used targets and attributes sets and their relationship to create the queries, check appendix A.

Finally, the sentence template string was added. As it was mentioned before the sentence template is a string that sets the grammar relationship between the targets and attributes. This field was generated manually by the researcher, and each one is considered the best and simplest grammatically correct sentence to connect the targets and attributes, which is another reason for the previous split (singular terms use ‘is’ and plural terms use ‘are’ as a word connection). Table 3 depicts the sentence templates added, the number of queries that use them, and some examples.

| Sentence Template | Queries |
|---|---|
| [TARGET] like [ATTRIBUTE] | Q1 and Q10 |
| [TARGET] likes [ATTRIBUTE] | Q11, Q20, Q21 and Q30 |

|  |  |
|---|---|
| [TARGET] are interested in [ATTRIBUTE] | Q2 and Q3 |
| [TARGET] is interested in [ATTRIBUTE] | Q12, Q13, Q22 and Q23 |
| [TARGET] are [ATTRIBUTE] | Q4, Q5, Q6, Q7, Q8, and Q9 |
| [TARGET] is [ATTRIBUTE] | Q14, Q15, Q16, Q17, Q18, Q19, Q24, Q25, Q26, Q27, Q28, Q29, Q31, Q32, Q33, Q34, Q35, Q36, Q37, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q45, Q46, Q47, Q48, Q49, Q50, Q51, Q52, Q53, and Q54 |

Table 3: Sentence Template used and some examples of queries where they are used.

For example, the queries with plural targets can not use the connection words "likes" and "is" because they need to use "like" and "are". Attributes that are related to fields and processions (math, art, career, etc) use the connection "like/likes" and "is interested in/are interested in", but sentimental concepts like 'positive, negative, pleasant, unpleasant, etc use the connection word 'is/are'.

The decision of choosing a sentence template that fits with a query was made manually considering the work of Kurita et al. (2019). Also, the chosen sentence template is considered by the researcher as the grammatically best option depending on the attributes.

These manual splits do not affect the logic of the queries, but since they are not the same queries as used in other experiments the results are not exactly the same (queries details in Appendix A.3).

### 3.1.3. Cleaning

As it was mentioned in the last section, the target and attributes were designed for a general fairness evaluation and not for this or similar experiments. There are two common problems when the word sets are not validating that could produce an unfair comparison between Static and Contextual Word Embeddings models:

- Out of Vocabulary Words.
- Grammatically Incorrect Sentences.

The Static Word Embeddings models have a vocabulary, a list of all words that can be transformed into a word embedding by the model when a word is not in this list, it should be ignored by the fairness evaluation. The WEFE framework counts this out of vocabulary words and throws an error when they exceed the configurable tolerance (20% by default).

This concept of out of vocabulary words does not apply to Contextual Word Embedding models. These models produce a word embedding for every word, if the word does not exist the closest meaning is returned as the word embedding.

In order to have a fair comparison between Static and Contextual Word Embeddings social bias, those words that are not included in our Static Word Embeddings models vocabularies (GloVe and Word2Vec) were excluded from the dataset (e.g. Einstein and NASA were removed for science terms set).

After the out of vocabulary words were removed, the sentence templates need to be validated. These sentence templates were designed to fit with the target-attribute relationship, but some attribute word sets are too big to be checked manually (between 5 and 3287 words), so a grammar validation was performed.

The language_tool_python library (Morris, 2020) is used to evaluate if a sentence (after the replacement of the target and attribute) is grammatically correct, some manual analysis is performed to ensure that the possible problems are with some attributes, e.g. positive word set has the same word with multiple conjugations like “contaminate”, “contaminated”, “contaminates”, “contaminating”, and “contamination”, “he is contaminated” is correct, but “he is contaminates” is not. Those words that produce a grammatically incorrect sentence were removed from the attribute word sets. Fortunately, after the cleaning no attribute word sets were empty, so it was not necessary to reduce the number of queries.

The positive and negative attribute word sets are special because they consist of 1154 and 3287 words respectively. These word sets are too big in comparison with the others (the mean of the number of words in a set excluding these two sets is 18 words). In order to reduce the computational cost of this experiment, those word sets were reduced to 115 words each by random sampling.

### 3.1.4. Result

The result is a 54 rows dataset (30 for Gender, 15 for Ethnicity, and 9 for Religion bias), each one represents a (2,2) template query (2 target sets and 2 attribute sets, see section 2.2.1). Also, each row has a sentence template, so either a query or a contextual query can be generated from it, so this query dataset can be used for Static and Contextual Word Embeddings pre-trained models. Table 4 shows the definition of the query dataset.

| Field | Description | Data Type | Values |
|---|---|---|---|
| qid | Query Identifier. | Numerical | 1 to 54 |
| type | Type of query (Gender, Religious, or Ethnicity). | Nominal | Gender, Religious, or Ethnicity |
| tname1 | Name of the first target. | String | Plural male terms |
| target1 | List of words that represent the first target. | Array | ['sons', 'fathers', 'men', 'boys', 'males', 'brothers', 'uncles', 'nephews'] |
| tname2 | Name of the second target. | String | Plural female terms |
| target2 | List of words that represent the second target. | Array | {'daughters', 'mothers', 'women', 'girls', 'females', 'sisters', 'aunts', 'nieces'} |
| aname1 | Name of the first attribute. | String | Math |
| attribute1 | List of words that represent the first attribute. | Array | {'math', 'algebra', 'geometry', 'calculus', 'equations', 'computation', 'numbers'} |

|  |  |  |  |
|---|---|---|---|
| aname2 | Name of the second attribute. | String | Arts |
| attribute2 | List of words that represent the second attribute. | Array | {'poetry', 'art', 'dance', 'literature', 'novel', 'symphony', 'drama'} |
| sentence_template | A sentence that defines the linguistic relation between target and attribute. | String | [TARGET] like [ATTRIBUTE] |

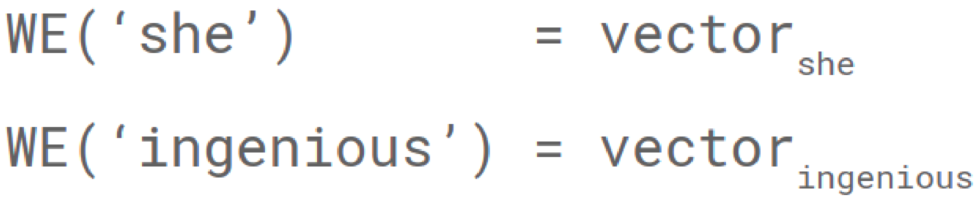
Table 4: Description of the Query Dataset.

The query dataset is useful to calculate the fairness metrics RND, RNSB and WEAT in the WEFE framework. The RND metric requires a (2,1) template query, but the WEFE framework internally can split a (2,2) query into two (2,1), calculate two RND metrics and get the mean, so the RND metric result can be compared with the other metrics.

This dataset can be used in other experiments using different metrics and word embedding pre-trained models, and it even can be extended to measure more types of social bias.

### 3.2. Getting Word Embeddings

Getting word embeddings from Static Word Embeddings pre-trained models is only a matter of passing the word to the model and getting the representation vector (see Figure 5). We want to get the equivalent in Contextual Word Embeddings pre-trained models, but for these models, the process is much more complicated.



```
WE('she')        = vector_she
WE('ingenious')  = vector_ingenious
```

Figure 5: Getting Word Embeddings in Static Word Embeddings models.

In a fairness evaluation, it is necessary to get the word embeddings of the target and attribute word sets. Using this process, we can define the T and A sets necessary to perform the fairness evaluation (see section 2.2.2). The following is the formal definition of these sets:

$$T_x = \{t_1, t_2, \ldots, t_n\}$$
$$A_x = \{a_1, a_2, \ldots, a_n\}$$

Where:

- $T_x$ is the xth set of word embeddings for the targets.
- $A_x$ is the xth set of word embeddings for the attributes.
- $t_x$ is a word embedding (vector) that represents the $x^{th}$ word in the target word set.
- $a_x$ is a word embedding (vector) that represents the $x^{th}$ word in the attribute word set.
- n is the number of vectors in the set.

For example, a (2,2) template query will produce the $T_1$, $T_2$, $A_1$, and $A_2$ sets of word embeddings, and these sets are required by the fairness metric formula.

As it was mentioned before Contextual Word Embeddings works with sentences instead of words, producing sentence embeddings. It is necessary to extract the word embeddings from the sentence embeddings. For this experiment two approaches were implemented:

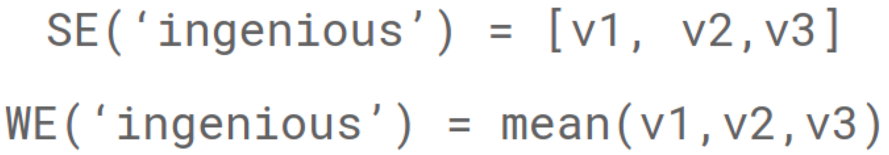
- Word Embeddings from Single Word Sentences
- Word Embeddings from Sentence Templates

The process of getting word embeddings from single word sentences uses a single word (target or attribute) as a sentence (e.g. ‘she’, ‘ingenious’) which is not too recommendable because no context can be extracted from the sentence.

Getting word embeddings from sentence templates needs the sentence template string mentioned in the query dataset to ensure we are using the desired context (e.g. 'she is ingenious'). These methods are explored in sections 3.2.1 and 3.2.2.

### 3.2.1. From Single Word Sentences

The Contextual Word Embeddings models do not have a vocabulary set, and some words can be represented by more than just one-word embedding (the quantity of words is the same for a word, what changes by the context is the values of these word embeddings). In order to have a single word embedding, all the produced word embeddings are combined using the mean (see Figure 6). Having a single word embedding representation helps to perform a fair comparison with word embedding from Static Word Embedding models.



```
SE('ingenious') = [v1, v2,v3]
WE('ingenious') = mean(v1,v2,v3)
```

Figure 6: Getting Word Embeddings from a Single Word Sentence.

Unfortunately, it is unclear what is the exact meaning of this vector. it could be a representation of the word without any context (exactly like the Static Word Embeddings) or the mean of the vectors produced by the word in multiple contexts. This approach is not mentioned or used in related work.

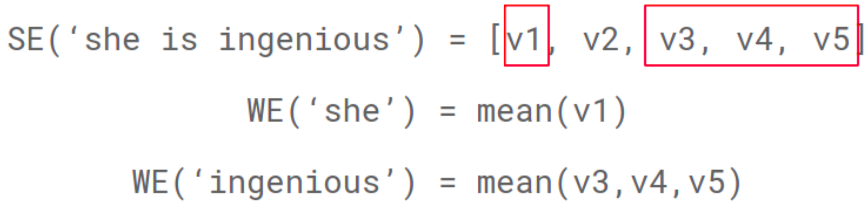
Due to the result of this method being similar to getting word embeddings in Static Word Embedding models, the formal definition for the T and A sets are the same as the last section.

### 3.2.2. From Sentence Templates

As it was explained in section 2.4.1, May et al. (2019) used sentences embeddings instead of word embeddings to measure WEAT, e.g. instead of using “she” and “ingenious” as a target and attribute word, they used the sentences “she is here” and “this is ingenious” to get the target and attribute vectors, but the calculated metric is considered SEAT (Sentence Embedding Association Test) instead of WEAT (Word Embedding Association Test). Even when the concept is similar they are not the same metric and they should not be compared.

Kurita et al. (2019) went beyond this idea and used a template sentence to set a context based on the relationship between the target and the attribute (e.g. “[TARGET] is a [ATTRIBUTE]”). Once they get the sentence embeddings, the word embeddings for each word are obtained from it.

This last approach sounds like the best option. Unfortunately, after checking their implementation, they did not consider words with more than one vector representation, so this approach is combined with a similar strategy of using the mean to get only one representation (see Figure 7).



```
SE(‘she is ingenious’) = [v1, v2, v3, v4, v5]
WE(‘she’) = mean(v1)
WE(‘ingenious’) = mean(v3,v4,v5)
```

Figure 7: Getting Word EMbeddings from a Sentence Template.

This approach seems to be the most accurate because it sets the exact context and each word generates one single word embedding representation.

As it was mentioned in section 2, some literature used PCA to show a 2D representation of the word embedding vectors to explain the fairness metrics. Figure 8 shows how this approach fits with the general idea of measuring bias using the distance between vectors (WEAT and RND). While in Static Word Embeddings there is only one representation per word, in Contextual Word Embeddings there are multiple representations even when the linguistic context is the same, e.g. in ‘she is ingenious’ and ‘he is ingenious’ the word ‘ingenious’ has a male and female representation even when the linguistic context is the same, but what matters in measuring bias is the distance, so even when the vectors are different we can consider bias if one target is closer to their attribute.

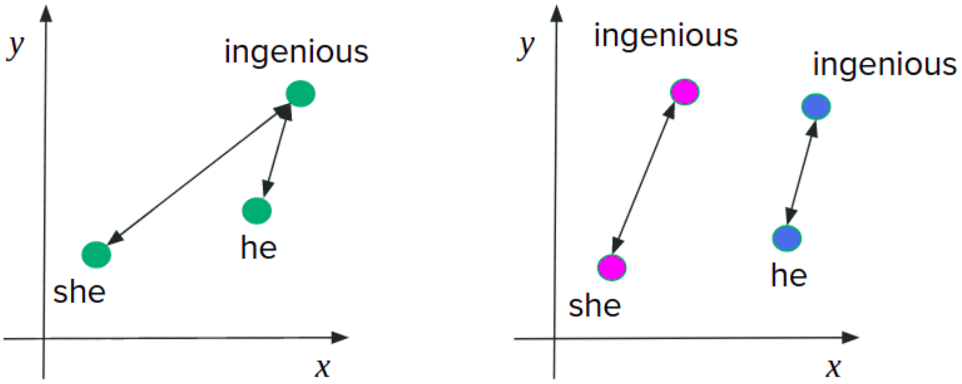


Figure 8: Static (left) and Contextual (right) Word Embeddings 2D representations. In both cases, there is a bias because he is closer to ingenious even when the representations are not the same (contextualised).

On the other hand, this approach forces a relation between the target and the attribute. In previous approaches, the targets are the same for both attributes, but in this one there is a set of targets for each set of attributes. This changes the previous definition of T and A.

For example, consider a (2,2) template query for male and female terms with science and art terms. Normally the targets and attributes are transformed to word embeddings separately, but now a relationship was set, so there are targets for the science terms and targets for the art terms, also there are attributes for the male terms and attributes for the female terms.

This includes a new term, class. The classes for targets are defined by the attributes and the classes for attributes are defined by the targets, so the number of classes is the same as the template query. Our (2,2) template query example has two classes of target and two classes of attributes. Targets 1 are the male terms, and target 1 class 0 are the male terms for science and target 1 class 1 are the male terms for art.

The following is the formal definition of T and A for this approach:

$$T^c_x = \{t^c_1, t^c_2, \ldots, t^c_n\}$$
$$A^c_x = \{a^c_1, a^c_2, \ldots, a^c_n\}$$

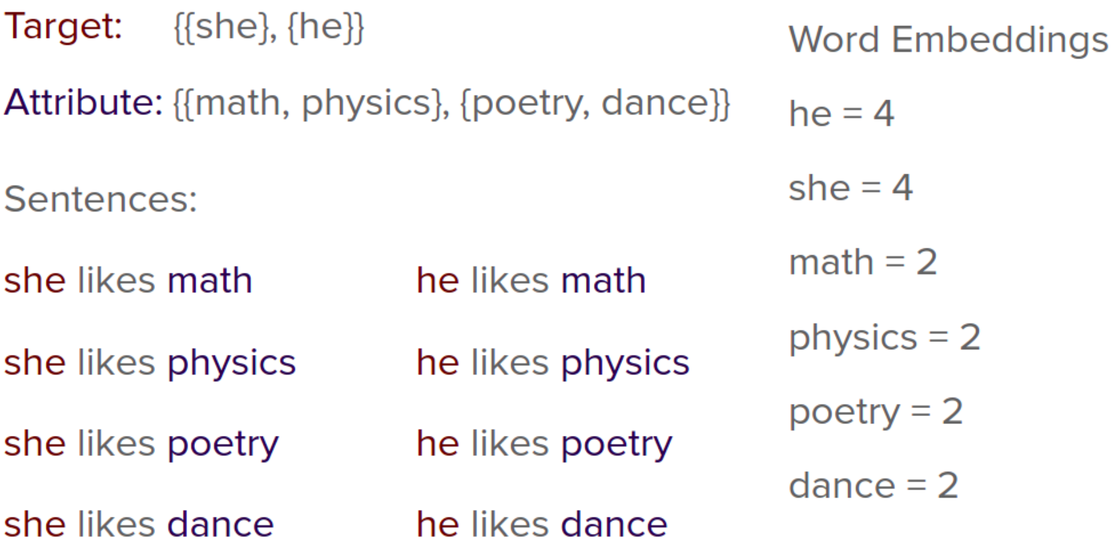
Where:

- $T^c_x$ is the xth target word embedding set of class c.
- $A^c_x$ is the xth attribute word embedding set of class c.
- c is the class of the attribute-target relationship specified by the query template.
- $t^c_x$ is a word embedding that represents the $x^{th}$ word target in class c.
- $a^c_x$ is a word embedding that represents the $x^{th}$ word attribute in class c.
- n is the number of vectors in the set.

In a (2,2) template query using the previous approach generates two targets and two attributes ($T_1$, $T_2$, $A_1$, and $A_2$), using this approach generates the double ($T^0_1$, $T^1_1$, $T^0_2$, $T^1_2$, $A^0_1$, $A^1_1$, $A^0_2$, and $A^1_2$). Using our example of the male and female terms with science and art terms query the Ts and As are the following:

- $T^0_1$ is the word embeddings of the male targets for science terms.
- $T^0_2$ is the word embeddings of the female targets for science terms.
- $T^1_1$ is the word embeddings of the male targets for art terms.
- $T^1_2$ is the word embeddings of the female targets for art terms.
- $A^0_1$ is the word embeddings of the science attributes for male terms.
- $A^0_2$ is the word embeddings of the science attributes for female terms.
- $A^1_1$ is the word embeddings of the art attributes for the male terms.
- $A^1_2$ is the word embeddings of the art attributes for the female terms.

Also, this relationship increases the number of targets and attributes, e.g. if the query has 4 attributes (2 for each class) there will be 4 representations for each target, also each attribute will have a single representation for each target. Figure 9 shows this process using a simple (2,2) template query, at the end, there will be a 4-word embedding representation for each target and a 2-word embedding representation for each attribute. It is important to mention that this is just an example, the RNSB metric requires a query to have at least 6 targets.



Target: {{she}, {he}}

Attribute: {{math, physics}, {poetry, dance}}

Sentences:

| | |
|---|---|
| she likes math | he likes math |
| she likes physics | he likes physics |
| she likes poetry | he likes poetry |
| she likes dance | he likes dance |

Word Embeddings

he = 4

she = 4

math = 2

physics = 2

poetry = 2

dance = 2

Figure 9: Example getting embeddings.

These changes on T and A affect the sources in the definition of the fairness metrics formula. Since the maths definition and concept is the same, new variants of the fairness metrics formulas for contextual word embeddings are needed to ensure the calculation of the metric follows the relation between each class. Section 3.3 explores in detail the necessary changes in the fairness metrics when using Contextual Word embedding models.

### 3.3. Contextual Fairness Metrics

When using the word embeddings from the sentence templates approach, there is a relation between targets and attributes. The term class is used to specify the relationship between targets and attributes (class 0 in targets means the first attributes while class 0 in attributes means the first targets). This forces the creation and usage of a new variant of the fairness metrics (RND, RNSB and WEAT) formulas where the source of the word embeddings is specified (which target and attribute set needs to be used depending on its class).

In all metrics, a combination of terms is needed in the targets or attributes word embeddings. This combination is the mean of all the same terms in a class, e.g. a target ‘she’ has multiple word embeddings representations (one per attribute), so the combination reduces these representations to just one. In this example, if the targets are combined there will be only a one-word embedding representation in each target class (e.g. one ‘she’ in T0 and another in T1). This combination is necessary to fulfil the metric requirements and have the same results.

The RND metric is a sum of the Euclidean Norm (see section 2.2.2) of each attribute with respect to all targets mean, so having repeated attributes affects this metric. That is why the combination of attribute terms is necessary, in the targets is not necessary because the formula requires all the targets average. The new variant definition of RND for contextual fairness evaluation is the following:

$$F_{RND}({T^0}_1, {T^0}_2, A^0, A^1) = \sum_{x1 \in A0,\ x2 \in A1} (\| avg({T^0}_1) - x1 \|_2 - \| avg({T^0}_2) - x2 \|_2)$$

Where:

- $F_{RND}$ is the RND fairness metric.
- ${T^c}_x$ is the xth target word embeddings vector set for class c (0/1).
- $A^c$ is the attribute word embeddings vector set for class c (0/1).
- $||•||_2$ is the Euclidean Norm function (see Figure 2).
- avg(•) is the averaging of all the values in a vector.

The WEAT metric is similar to the RND, but the process is the opposite. It is a sum of the Cosine Similarity (see section 2.2.2) of each target with respect to its attributes, so the same combination of terms is required, but this time in the targets. The new variant definition of WEAT for contextual fairness evaluation is the following:

$$F_{WEAT}(T^0_1, T^1_1, T^0_2, T^1_2, A^0_1, A^1_1, A^0_2, A^1_2) = \sum_{w1 \in T01, w2 \in T11} d(w1, w2, A^0_1, A^0_2) - \sum_{w1 \in T02, w2 \in T12} d(w1, w2, A^1_1, A^1_2)$$

$$d(w1, w2, A_1, A_2) = (\ \mathrm{mean}_{x \in A1} \cos(w1, x)\ ) - (\ \mathrm{mean}_{x \in A2} \cos(w2, x)\ )$$

Where:

- $F_{WEAT}$ is the WEAT fairness metric.
- $T^c_x$ is the xth target word embeddings vector set for class c (0/1).
- $A^c_x$ is the xth attribute word embeddings vector set for class c (0/1).
- cos(•,•) is the cosine similarity function (see Figure 2).

The RNSB metric uses the KL divergence of the probability distribution of the targets for a classifier model trained with the attributes and the uniform distribution (see section 2.2.2). The combination of terms required for this metric is for both targets and attributes. The new variant definition of RNSB for contextual fairness evaluation is the following:

$$NP = w \in T^0_1 \cup T^1_1 \cup T^0_2 \cup T^1_2 \ \ (C_{(A01 \cup A11, A02 \cup A12)}(w))$$

$$P = NP \,/\, \sum_{x \in X} NP(x)$$

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

Where:

- $F_{RNSB}$ is the RNSB fairness metric.
- $T^c_x$ is the xth target word embeddings vector set for class c (0/1).
- $A^c_x$ is the xth attribute word embeddings vector set for class c (0/1).

- $C_{(A1,\ A2)}(\bullet)$ is a binary classifier trained with $A_1$ for negative class, and $A_2$ for positive class.
- $D_{KL}$ is an LK Divergence (see Figure 2).
- NP is the negative probability distribution of the targets.
- P is the normalised negative probability distribution of the targets, $\sum P(w) = 1$.
- U is the Uniform Distribution.

All these variant formulas were tested using this approach in Static Word Embeddings and the results were the same as the original formulas just like was expected.

### 3.4. Ranking

The Fairness Evaluations in Static and Contextual Word Embeddings produce a set with the metric values for each model, bias type, and fairness metric. Over these values, a ranking test is performed to get the rankings by Metric and Type of Bias (see Table 5).

| Field | Description | Data Type | Values |
|---|---|---|---|
| model | Name of the pre-trained model. | Nominal | GloVe, Word2Vec, BERT or RoBERTa |
| type | Name of the type of bias measure. | Nominal | Gender, Religious, or Ethnicity |
| metric | Name of the fairness metric. | Nominal | RND, RNSB or WEAT |
| value | Values generated by the fairness evaluation. The absolute mean of the results of all queries. | Numerical | 0.0 to 1.0 |
| ranking | Ranking of the value by metric and bias type. | Numerical | 0.0 to 4.0 |

Table 5: Description of the Result dataset.

This ranking is based on the Badilla et al. (2020) experiment, easy comparison between different metrics. The rankings are calculated using the rank implementation (method='first') of the Pandas library.

In addition to the three social bias types, the mean of all the results was processed and tagged as the Overall bias type. This was made with the intention of having a general (or overall) view of the results.

In the complete experiment, two result sets are generated, one using the word embeddings from the single word sentences approach and the other using the word embeddings from the sentence templates approach. Each dataset requires a ranking process and they can be considered separate experiments.

### 3.5. Technical Challenges

As was mentioned before, the WEFE framework was not designed to process Contextual Word Embedding pre-trained models, so it was modified for this experiment. The WEFE framework is an open-source project available on GitHub (https://github.com/dccuchile/wefe), so the code was downloaded, modified, tested, and prepare for a correct integration to the main project (https://github.com/dccuchile/wefe/pull/25).

After some meetings with the WEFE Developer Team, we decided to prepare a specific branch for this change and increase the scope of this sub-project. The idea is to prepare the WEFE framework to accept more Contextual World Embedding pre-trained models like ELMo and support different approaches to get word embeddings from Contextual World Embeddings.

Anyway, the code for this experiment is functional and available online. The WEFE framework is developed in Python and this experiment was performed in a Google Collab Notebook. The following are the necessary and more important changes implemented during this dissertation project.

### 3.5.1 The WEFE Development Team

The WEFE developer team is headed by Pablo Badilla and Felipe Bravo-Marque from the Department of Computer Science, Universidad de Chile.

The interaction with this team was by emails with Pablo Badilla to discuss the approaches and necessary changes in the framework. Then I joined the weekly meetings to discuss the experiments that were being developed and future work. These meetings were online in the discord channel of the WEFE Developer Team.

### 3.5.2. Word Embedding Container

The WEFE framework internals includes a Word Embedding model container class. This class is in charge of performing different operations on a Gensim (Rehurek & Sojka, 2011) Word Embedding model. All the functions and utils modules are prepared to receive a WordEmbeddingModel object otherwise they throw an exception error.

Unfortunately, the Gensim library does not include Contextual Word Embedding models like BERT, ELMo, or RoBERTa, so a new Word Embedding model container class should be created and this class needs to be accepted by the rest of the framework. This problem was solved using an inheritance hierarchy (see Figure 10)  where the current container and the new one have the same parent. The WEFE framework accepts and processes the container classes as their parent using polymorphism.

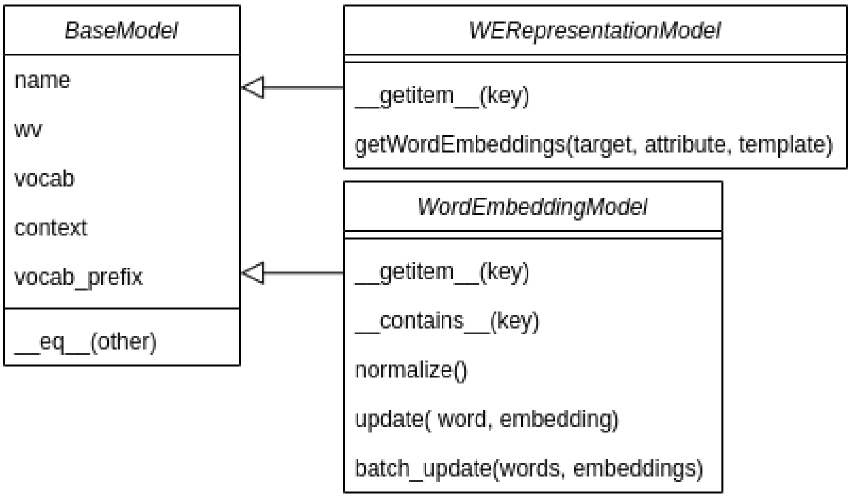


Figure 10: Word Embedding Container Class Diagram.

The new container, called WERepresentationModel, works with a RepresentationModel from the simpletransformers library (Rajapakse, 2020). This library makes it easy to download Contextual Word Embedding pre-trained models based on transformers such as BERT and RoBERTa, also the combination strategy of Sentence Embeddings is configurable, easing the process of getting word embeddings from sentence embeddings.

The approaches of sections 3.2.1 and 3.2.2 describe the correct extraction of word embeddings from sentence embeddings. These extractions were implemented in the WERepresentationModel where only BERT based models are available.

The implementation of those approaches involves the mean of the word embeddings vectors generated from each word. The BERT based models generate two extra tokens CLS at the beginning of the sentence and SEP at the end of the sentence, those extra tokens were ignored to avoid noise in the word embedding vectors.

To calculate the mean of the word embeddings vectors is necessary to know the number of vectors generated from a word, in the getting word embeddings from the sentence templates approach, it is difficult to know this, so first, the tokens for each word is obtained and the mean is calculated based on the number of tokens from a word.

### 3.5.3. Embeddings Set Container

The WEFE framework internals offers a variety of well designed and implemented tools to perform fairness evaluation and extend its functionality to add new metrics and experiment design. Unfortunately, one of the weakest parts of the WEFE internals is the object to store the word embeddings obtained from a query.

This object called EmbeddingSets, was not designed to store more information about the word embedding such as class, or relationships. That is why a new Embeddings Set Container was implemented simulating a dataset where a target-attribute relationship is stored with their respective class and word embedding vector (see Table 6).

| Field | Description | Data Type | Values |
|---|---|---|---|
| target | Related target word, | String | ‘she’, ‘he’, etc. |
| attribute | Related attribute word. | String | ‘Ingenious’, ‘ugly’, etc. |
| tclass | Class of the target. | Numerical | 1, 2, 3, etc. |
| aclass | Class of the attribute. | Numerical | 1, 2, 3, etc. |
| tvector | Vector or word embedding of the target. | Array | [-0.397, ..., 0.334] |
| avector | Vector or word embedding of the attribute. | Array | [-0.397, ..., 0.334] |

Table 6: Description of the Embeddings Set Container.

Thanks to this new container it is possible to get the word embeddings from the queries using the sentence embedding approach, and it is used for each metric implementation to calculate the needed fairness metric.

### 3.6. Conclusion

The goal of this research is to perform a fair comparison between Static and Contextual Word Embeddings. Based on the literature review, the sentence embedding approach was used to calculate the fairness metrics and the ranking test to compare the results.

This experiment was executed twice, using different approaches to get the word embeddings from the Contextual Word Embeddings pre-trained models. The results were two result sets and the conclusions are from both sets.

It would be interesting to do more research on getting word embeddings vectors from Contextual Word Embeddings pre-trained models. The literature review supports the word embeddings from the sentence templates approach, but there is no previous work that uses the word embeddings from the single sentences approach. There is no real evaluation of what represents the resulting word embedding using single word sentences (the mean of all the possible representations/context or an un-contextualized representation). The resulting word embeddings of these and other possible techniques could be compared and evaluated, but this is beyond the purpose of this dissertation work.

## 4. RESULTS

Using the query dataset of section 3.1 and the ranking process of section 3.4 two experiments were executed, each one using a different approach to get word embeddings in Contextual Word Embeddings pre-trained models (getting word embeddings from single word sentences and getting word embeddings from template sentences).

Table 7 shows the results of the executed experiments using both approaches. The results are organised in gender, ethnicity and religion social bias, also the mean of those metrics is included labelled as overall. Each cell shows the ranking and the fairness metric absolute value is in parenthesis. The results in italic font are for the word embeddings from the single word sentences approach, while the results with normal font are for the word embeddings from the sentence template approach. In GloVe and Word2Vec, fairness metric values are the same in both approaches because the approaches only apply for Contextual Word Embeddings pre-trained models, but the rankings are not necessarily the same.

| Model | Gender | | | Ethnicity | | |
|---|---|---|---|---|---|---|
| | RND | RNSB | WEAT | RND | RNSB | WEAT |
| GloVe | 3 (0.522) | 2 (0.045) | 3 (0.404) | 2 (0.313) | 2 (0.090) | 4 (0.637) |
| | *3 (0.522)* | *2 (0.045)* | *3 (0.404)* | *2 (0.313)* | *2 (0.089)* | *4 (0.637)* |
| Word2Vec | **1 (0.189)** | **1 (0.021)** | 4 (0.843) | **1 (0.071)** | **1 (0.013)** | 3 (0.342) |
| | ***1 (0.189)*** | ***1 (0.021)*** | *4 (0.843)* | ***1 (0.071)*** | ***1 (0.013)*** | *3 (0.342)* |
| BERT | 2 (0.029) | 3 (0.149) | 2 (0.178) | 4 (0.508) | 3 (0.481) | 2 (0.125) |
| | *4 (0.626)* | *3 (0.066)* | *2 (0.149)* | *4 (0.927)* | *3 (0.126)* | *2 (0.26)* |
| RoBERTa | 4 (0.644) | 4 (0.264) | **1 (0.059)** | 3 (0.445) | 4 (0.491) | **1 (0.036)** |
| | *2 (0.304)* | *4 (0.1)* | ***1 (0.069)*** | *3 (0.434)* | *4 (0.182)* | ***1 (0.113)*** |
| Model | Religion | | | Overall | | |
| | RND | RNSB | WEAT | RND | RNSB | WEAT |
| GloVe | 2 (0.139) | 2 (0.047) | 4 (1.367) | 2 (0.325) | 2 (0.061) | 4 (0.803) |
| | *2 (0.139)* | *2 (0.047)* | *4 (1.367)* | *2 (0.325)* | *2 (0.06)* | *4 (0.803)* |

| | | | | | | |
|---|---|---|---|---|---|---|
| Word2Vec | **1 (0.065)** | **1 (0.006)** | 3 (0.833) | **1 (0.108)** | **1 (0.014)** | 3 (0.673) |
| | ***1 (0.065)*** | ***1 (0.006)*** | *3 (0.833)* | ***1 (0.108)*** | ***1 (0.014)*** | *3 (0.673)* |
| BERT | 3 (0.175) | 3 (0.127) | 2 (0.229) | 3 (0.327) | 3 (0.252) | 2 (0.177) |
| | *3 (0.249)* | *4 (0.066)* | *2 (0.449)* | *4 (0.601)* | *3 (0.086)* | *2 (0.286)* |
| RoBERTa | 4 (0.364) | 4 (0.157) | **1 (0.048)** | 4 (0.484) | 4 (0.304) | **1 (0.048)** |
| | *4 (0.507)* | *3 (0.063)* | ***1 (0.108)*** | *3 (0.415)* | *4 (0.115)* | ***1 (0.097)*** |

Table 7: Ranking and absolute fairness metrics (RND, RNSB and WEAT) values resulted from measuring different social bias types in word embedding models. In normal font are the results using the getting word embeddings from sentence template approach and in italic font are the results using the getting word embeddings from single word sentences. In all cases the best result is shown in bold.

The cells with Bold font are the ones with the best results. It is important to mention that the pattern is the same for both approaches even when the metric values are not. There is a contrast between fairness metrics; the results in WEAT are almost the opposite as the results of RND and RNSB. Later in this section, we will see that these two metrics are strongly correlated.

Taking advantage of the ranking values, figure 11 shows the sum of ranks in each model by social bias and the getting word embedding approach. The superiority of the Word2Vec model is visible, but there is no similarity between the results of both approaches. The WEAT metric favours the RoBERTa model, being the one with the lower level of social bias in all social bias types and in both approaches.

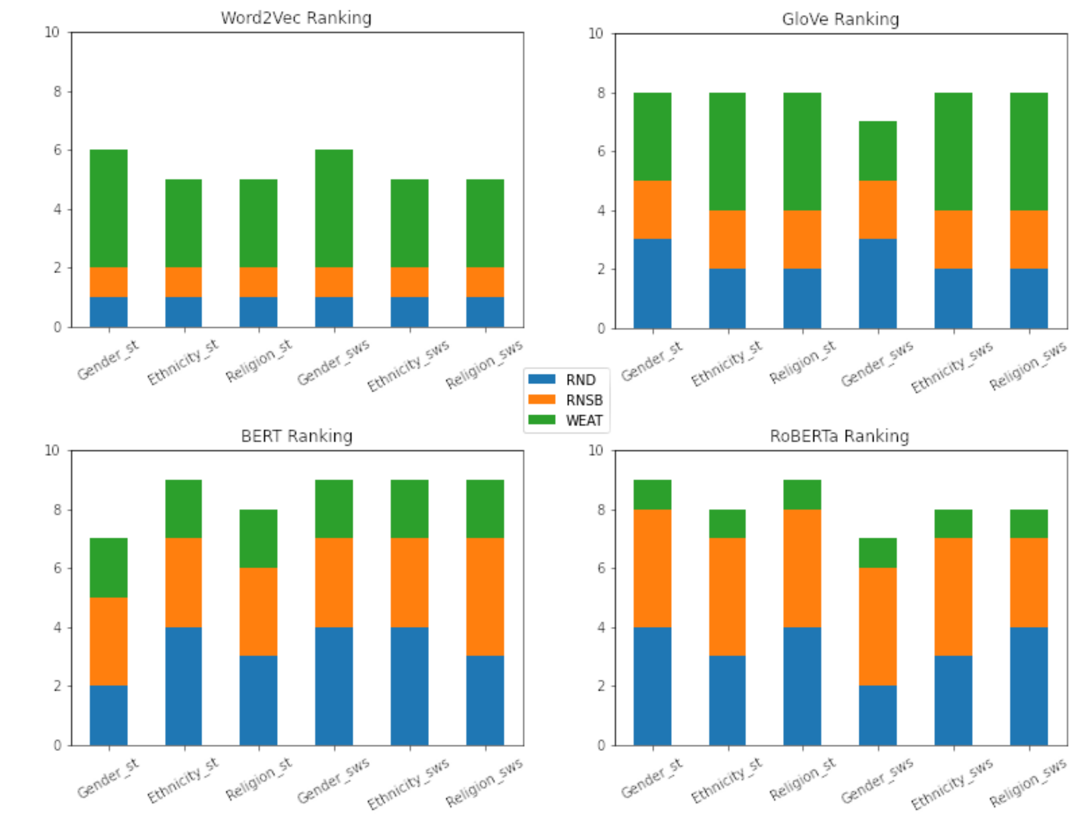


Figure 11: Ranking results by model. The Gender, ethnicity and religion bias rankings for the sentence template (ST) and single word sentence (SWS) approaches are included.

Figure 12 also shows the ranking values of both approaches, but in this case, is the overall (the mean) of the ranking values. Using this overall it is clear that the Contextual Word Embeddings pre-trained models do not have a lower level of gender, ethnicity and religion bias than the Static Word Embeddings pre-trained models. The same conclusion is for both approaches using the RND, RNSB and WEAT fairness metrics.

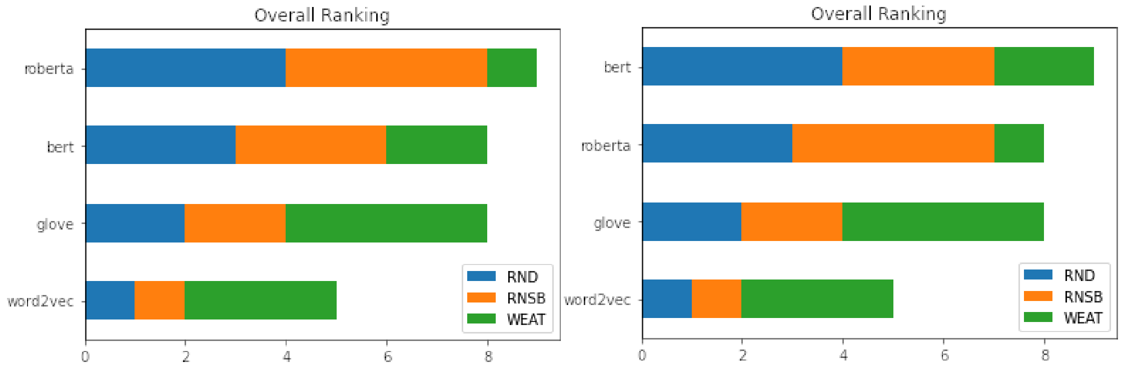
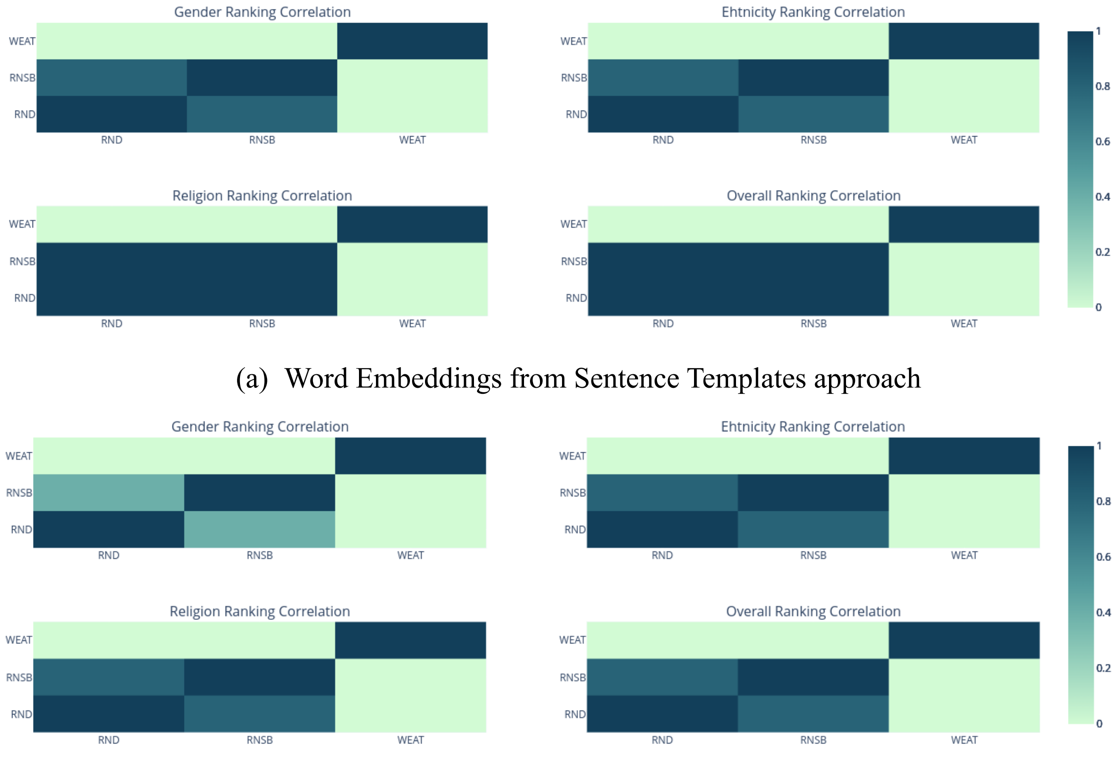


Figure 12: Overall ranking values of each model. On the left, the results use the word embeddings from the template sentence approach and on the right, the results use the word embeddings from the single word sentences.

Finally, figure 13 shows the Spearman correlation matrix for the rankings of all fairness metrics in both approaches (including the overall). The pattern is the same in both approaches; the RND and RNSB metrics are strongly correlated (stronger using the word embeddings from sentences template approach than using the word embeddings from single word sentences approach). In Badilla et al. (2020) the correlation between RND and RNSB is much stronger than the correlation between these metrics and WEAT, but this difference is not as clear as in this work. This could be normal considering that the queries are not the same.



(a) Word Embeddings from Sentence Templates approach

(b) Word Embeddings from Single Word Sentences approach

Figure 13: Spearman correlation matrix of rankings by different fairness metrics (RND, RNSB and WEAT). Each image shows the results using a specific approach.

In conclusion, measuring the RND and RNSB fairness metrics in Contextual and Static Word Embeddings pre-trained models it is clear that the contextual models BERT and RoBERTa contain a higher level of social bias than the static ones GloVe and Word2Vec, but if the WEAT is the fairness metrics the results are the opposite.

# 5. DISCUSSION AND FUTURE WORK

The main contributions of this dissertation work are:

1. The query dataset, a dataset with the necessary information(extracted from previous research work) to create a set of Gender, Ethnicity and Religion regular queries and contextual queries.
2. The Contextual Query, a new variant of the query proposed by Badilla et al. (2020) that can be used to perform a fairness evaluation over Contextual Word Embeddings.
3. The sentence templates approach, an approach to get word embedding from Contextual Word Embeddings pre-trained models is the extension of the work of Kurita et al. (2020).
4. The comparison of RND, RNSB and WEAT fairness metric of Static (Word2Vec and GloVe) and Contextual (BERT and RoBERTa) Word Embeddings pre-trained models. Surprisingly, overall the Contextual Word Embeddings models contain a higher level of social bias than Static Word Embeddings models. The RND and RNSB ranks are strongly correlated while they are weakly correlated with WEAT.

The query dataset is a new dataset with the necessary targets, attributes and relationships to generate 54 standard or contextual queries. Most of the experiments for literature review use a small number of queries (some of them use around 5 queries) and the source datasets are not designed specifically to measure social bias in word embeddings. This proposed query dataset could be a standard for this kind of experiment.

A possible problem of the query dataset is the static sentence template, each query includes a sentence template, this sentence template was chosen to set a grammatically correct relationship between the targets and attributes terms, but some other approaches decide to set this sentence template by random. If the researcher wants to use the sentence templates by random the query dataset would need some modifications.

Surprisingly, the ranking results of the two methods to get word embeddings from pre-trained models are similar. The assumption was that getting word embeddings from the sentence templates method is much more accurate than getting word embeddings from the single word sentences method because the first one is based on literature review, involves more process, and takes much more execution time (12.5 hours vs 1 hour executing). Future research could be about which approach produces more accurate results.

The conclusions of Basta et al. (2019) and Kurita et al. (2020) are that contextualised word embeddings vectors have a lower level of social bias than the static ones, but their fairness evaluations use WEAT and Direct Bias. These fairness metrics use cosine similarity. Due to the not statistically significant results of May et al. (2019), they contemplate the possibility that cosine similarity is an inadequate measure of text similarity for sentence word embeddings for ELMo. Kurita et al. (2020) got the same results, WEAT for BERT fails to find any statistically significant social biases ($p < 0.01$), so they conclude that WEAT is not an effective measure for bias in BERT word embeddings.

Unlike RND and RNSB results, the WEAT results of this experiment show that contextualised word embeddings vectors have a lower level of social bias than the static ones. Unfortunately, the WEFE framework does not calculate the p-values, so we can not validate them. It is a possibility that the results of RND and RNSB fairness metrics produce statistically significant values and that could be a reason for the weak correlation and different conclusions between these metrics and WEAT.

Future work could involve the addition of p-values in WEFE and exclude those statistically not significant results without affecting the fair comparison or a comparison of p-values between fairness metrics in order to decide which one is more effective to measure social bias in contextual word embeddings.

# REFERENCES

Badilla, P., Bravo-Marquez, F., & Pérez, J. (2020). WEFE: The Word Embeddings Fairness Evaluation Framework. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 29, 430–436. https://doi.org/10.24963/ijcai.2020/60

Basta, C., Costa-jussà, M. R., & Casas, N. (2019). Evaluating the Underlying Gender Bias in Contextualized Word Embeddings. Proceedings of the First Workshop on Gender Bias in Natural Language Processing, 1, 33–39. https://doi.org/10.18653/v1/w19-3805

Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. Proceedings of the 30th International Conference on Neural Information Processing Systems, 4356–4364. https://dl.acm.org/doi/10.5555/3157382.3157584

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. Science, 356(6334), 183–186. https://doi.org/10.1126/science.aal4230

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., & Robinson, T. (2013). One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. ArXiv Preprint. https://arxiv.org/abs/1312.3005

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

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of

Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 4171–4186. https://doi.org/10.18653/v1/n19-1423

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. Proceedings of the National Academy of Sciences, 115(16), E3635–E3644. https://doi.org/10.1073/pnas.1720347115

Gonen, H., & Goldberg, Y. (2019). Lipstick on a Pig: Proceedings of the 2019 Conference of the North, 609–614. https://doi.org/10.18653/v1/n19-1061

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '04, 168–177. https://doi.org/10.1145/1014052.1014073

IBM Cloud Education. (2021, August 17). Natural Language Processing (NLP). IBM. https://www.ibm.com/cloud/learn/natural-language-processing

Kumar, V., Bhotia, T., & Kumar, V. (2020a). Fair Embedding Engine: A Library for Analyzing and Mitigating Gender Bias in Word Embeddings. Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS). https://doi.org/10.18653/v1/2020.nlposs-1.5

Kumar, V., Bhotia, T. S., Kumar, V., & Chakraborty, T. (2020b). Nurse is Closer to Woman than Surgeon? Mitigating Gender-Biased Proximities in Word Embeddings. Transactions of the Association for Computational Linguistics, 8, 486–503. https://doi.org/10.1162/tacl_a_00327

Kurita, K., Vyas, N., Pareek, A., Black, A. W., & Tsvetkov, Y. (2019). Measuring Bias in Contextualized Word Representations. Proceedings of the First Workshop on Gender Bias in Natural Language Processing, 1, 166–172. https://doi.org/10.18653/v1/w19-3823

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. International Conference on Learning Representations. https://openreview.net/forum?id=H1eA7AEtvS

Lee, K., He, L., & Zettlemoyer, L. (2018). Higher-order Coreference Resolution with Coarse-to-fine Inference. ArXiv Preprint. https://arxiv.org/abs/1804.05392

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv. Published. https://arxiv.org/abs/1907.11692

Manzini, T., Yao Chong, L., Black, A. W., & Tsvetkov, Y. (2019). Black is to Criminal as Caucasian is to Police: Detecting and Removing Multiclass Bias in Word Embeddings. Proceedings of the 2019 Conference of the North, 615–621. https://doi.org/10.18653/v1/n19-1062

May, C., Wang, A., Bordia, S., Bowman, S. R., & Rudinger, R. (2019). On Measuring Social Biases in Sentence Encoders. Proceedings of the 2019 Conference of the North, 622–628. https://doi.org/10.18653/v1/n19-1063

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. ArXiv Preprint, 1–12. https://arxiv.org/abs/1301.3781v3

Morris, J. (2020). language-tool-python. PyPI. Retrieved December 8, 2021, from https://pypi.org/project/language-tool-python/

Oscar Deho, B., William Agangiba, A., Felix Aryeh, L., & Jeffery Ansah, A. (2018). Sentiment Analysis with Word Embedding. 2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST). https://doi.org/10.1109/icastech.2018.8506717

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2227–2237. https://doi.org/10.18653/v1/n18-1202

Rajapakse, T. (2020). Simple Transformers. Simpletransformers. Retrieved December 9, 2021, from https://simpletransformers.ai/

Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2).

Sweeney, C., & Najafian, M. (2019). A Transparent Framework for Evaluating Unintended Demographic Bias in Word Embeddings. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 57, 1662–1667. https://doi.org/10.18653/v1/p19-1162

Wang, T., Lin, X. V., Rajani, N. F., McCann, B., Ordonez, V., & Xiong, C. (2020).

Double-Hard Debias: Tailoring Word Embeddings for Gender Bias Mitigation. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.484

Yang, Z., & Feng, J. (2020). A Causal Inference Method for Reducing Gender Bias in Word Embedding Relations. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 9434–9441. https://doi.org/10.1609/aaai.v34i05.6486

Zhao, J., Wang, T., Yatskar, M., Cotterell, R., Ordonez, V., & Chang, K. W. (2019). Gender Bias in Contextualized Word Embeddings. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 629–634. https://doi.org/10.18653/v1/n19-1064

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. W. (2017). Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2979–2989. https://doi.org/10.18653/v1/d17-1323

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. W. (2018). Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 15–20. https://doi.org/10.18653/v1/n18-2003

# APPENDIX A

This section contains the complete information of the query dataset.

### A.1 Targets Word Sets

| Id | Target Terms | Examples |
|---|---|---|
| T1 | Plural Male | 'sons', 'fathers', 'men', 'boys', 'males', 'brothers', 'uncles', 'nephews' |
| T2 | Single Male | 'male', 'man', 'boy', 'he', 'himself', 'him', 'his' |
| T3 | Single Male 2 | 'brother', 'father', 'uncle', 'grandfather', 'nephew', 'son' |
| T4 | Plural Female | 'daughters', 'mothers', 'women', 'girls', 'females', 'sisters', 'aunts', 'nieces' |
| T5 | Single Female | 'female', 'woman', 'girl', 'she', 'her', 'herself', 'hers' |
| T6 | Single Female 2 | 'sister', 'mother', 'aunt', 'grandmother', 'daughter', 'niece' |
| T7 | White last names | 'harris', 'nelson', 'robinson', 'thompson', 'moore', 'wright', 'anderson', 'clark', 'jackson', 'taylor', 'scott', 'davis', 'allen', 'adams', 'lewis', 'williams', 'wilson', 'martin', 'johnson' |
| T8 | Black last names | 'harris', 'robinson', 'howard', 'thompson', 'moore', 'wright', 'anderson', 'clark', 'jackson', 'taylor', 'scott', 'davis', 'allen', 'adams', 'lewis', 'williams', 'wilson', 'martin', 'johnson' |
| T9 | Hispanic last names | 'castillo', 'gomez', 'soto', 'gonzalez', 'sanchez', 'rivera', 'martinez', 'torres', 'rodriguez', 'perez', 'lopez', 'medina', 'diaz', 'garcia', 'castro', 'cruz' |
| T10 | Chinese last names | 'chung', 'liu', 'wong', 'huang', 'ng', 'hu', 'chu', 'chen', 'lin', 'liang', 'wang', 'wu', 'yang', 'tang', 'chang', 'hong', 'li' |
| T11 | Asian last names | 'cho', 'wong', 'tang', 'huang', 'chu', 'chung', 'ng', 'wu', 'liu', 'chen', 'lin', 'yang', 'kim', 'chang', 'shah', 'wang', 'li', 'khan' |
| T12 | Christianity | 'bible', 'christian', 'christianity', 'church', 'priest', 'jesus' |
| T13 | Islam | 'imam', 'islam', 'mosque', 'muslim', 'quran', 'muhammad' |
| T14 | Judaism | 'jew', 'judaism', 'rabbi', 'synagogue', 'torah' |
| T15 | Jew | 'jew', 'judaism', 'rabbi', 'synagogue', 'torah' |

### A.2 Attributes Word Sets

| Id | Attribute Terms | Examples |
|---|---|---|
| A1 | Career | 'executive', 'management', 'professional', 'corporation', 'salary', 'office', 'business', 'career' |
| A2 | Family | 'home', 'parents', 'children', 'family', 'cousins', 'marriage', 'wedding', 'relatives' |
| A3 | Science | 'science', 'technology', 'physics', 'chemistry', 'experiment', 'astronomy' |

| | | |
|---|---|---|
| A4 | Arts | 'poetry', 'art', 'dance', 'literature', 'novel', 'symphony', 'drama', 'sculpture' |
| A5 | Math | 'math', 'algebra', 'geometry', 'calculus', 'equations', 'computation', 'numbers', 'addition' |
| A6 | Intelligence | 'precocious', 'resourceful', 'inquisitive', 'inventive', 'astute', 'adaptable', 'reflective', 'discerning', 'intuitive', 'inquiring', 'judicious', 'analytical', 'luminous', 'venerable', 'imaginative', 'shrewd', 'thoughtful', 'sage', 'smart', 'ingenious', 'clever', 'brilliant', 'logical', 'intelligent', 'apt', 'genius', 'wise' |
| A7 | Appearance | 'alluring', 'voluptuous', 'blushing', 'homely', 'plump', 'sensual', 'gorgeous', 'slim', 'bald', 'athletic', 'fashionable', 'stout', 'ugly', 'muscular', 'slender', 'feeble', 'handsome', 'healthy', 'attractive', 'fat', 'weak', 'thin', 'pretty', 'beautiful', 'strong' |
| A8 | Sensitive | 'inhibited', 'complacent', 'sensitive', 'mellow', 'solemn', 'studious', 'intelligent', 'brilliant', 'rational', 'serious', 'contemplative', 'cowardly', 'timid', 'shy', 'passive', 'delicate', 'gentle', 'soft', 'quiet', 'working' |
| A9 | Man Roles | 'manager', 'executive', 'doctor', 'lawyer', 'programmer', 'scientist', 'soldier', 'supervisor', 'rancher', 'janitor', 'firefighter', 'officer' |
| A10 | Woman Roles | 'secretary', 'nurse', 'clerk', 'artist', 'homemaker', 'dancer', 'singer', 'librarian', 'maid', 'hairdresser', 'stylist', 'receptionist', 'counselor' |
| A11 | Man Occupations | 'physician', 'doctor', 'laborer', 'conservationist', 'proprietor', 'operator', 'mechanic', 'surveyor', 'physicist', 'machinist', 'architect', 'photographer', 'optometrist', 'millwright', 'sales', 'smith', 'manager', 'statistician', 'athlete', 'bailiff', 'clerk', 'boilermaker', 'official', 'conductor', 'porter', 'chemist', 'inspector', 'professor', 'salesperson', 'lawyer', 'farmer', 'electrician', 'sailor', 'geologist', 'setter', 'lumberjack', 'instructor', 'plasterer', 'judge', 'sheriff', 'surgeon' |
| A12 | Woman Occupations | 'cashier', 'bartender', 'teacher', 'baker', 'dancer', 'nutritionist', 'broker', 'dietitian', 'author', 'entertainer', 'economist', 'nurse', 'secretary', 'clerical', 'therapist', 'technician', 'veterinarian', 'attendant', 'janitor', 'weaver', 'musician', 'waitstaff', 'psychologist', 'designer', 'pharmacist', 'librarian', 'accountant', 'administrator', 'housekeeper' |
| A13 | Weapons | 'arrow', 'club', 'gun', 'missile', 'spear', 'dagger', 'harpoon', 'pistol', 'sword', 'blade', 'dynamite', 'hatchet', 'rifle', 'tank', 'bomb', 'firearm', 'knife', 'shotgun', 'teargas', 'cannon', 'grenade', 'mace', 'slingshot', 'whip' |
| A14 | Instruments | 'bagpipe', 'cello', 'guitar', 'lute', 'trombone', 'banjo', 'clarinet', 'harmonica', 'mandolin', 'trumpet', 'bassoon', 'drum', 'harp', 'oboe', 'tuba', 'bell', 'fiddle', 'harpsichord', 'piano', 'viola', |

| | | |
|---|---|---|
| | | 'bongo', 'flute', 'horn', 'saxophone', 'violin' |
| A15 | Occupations black | 'farmer', 'veterinarian', 'pilot', 'optometrist', 'physicist', 'dentist', 'chiropractor', 'geologist', 'statistician', 'plasterer' |
| A16 | Occupations Asian | 'plasterer', 'conductor', 'boilermaker', 'millwright', 'mason', 'conservationist', 'setter', 'plumber' |
| A17 | Occupations Hispanic | 'optometrist', 'veterinarian', 'physicist', 'geologist', 'pharmacist', 'chiropractor', 'statistician', 'millwright', 'setter' |
| A18 | Occupations white | 'plasterer', 'janitor', 'porter', 'cook', 'shoemaker', 'laborer', 'guard', 'baker', 'gardener' |
| A19 | Positive words | 'optimism', 'awarded', 'rightness', 'eminence', 'originality', 'pampers', 'flashy', 'pretty', 'blameless', 'groundbreaking', 'elation', 'cheerful', 'earnest', 'detachable', 'resounding', 'uncomplicated', 'won', 'creative', 'precious', 'calm', 'integral', 'genius', 'desiring', 'progressive', 'adaptive', 'beauty', 'luckiest', 'gutsy', 'majestic', 'promise', 'breathtaking' |
| A20 | Negative words | 'ugliness', 'weakness', 'revolting', 'hostility', 'naive', 'murky', 'mischievous', 'naïve', 'carelessness', 'catastrophe', 'vagueness', 'dirtbags', 'mischief', 'snobbish', 'worsening', 'flairs', 'invisible', 'delay', 'allergy', 'horrifying', 'oppressive', 'assassin', 'disgruntled', 'excuse', 'vain', 'excessive', 'baffling', 'kills', 'spinster', 'languid', 'allegations', 'deteriorating', 'failure', 'ruffian', 'distortion', 'diabolic' |
| A21 | Pleasant | 'caress', 'freedom', 'health', 'love', 'peace', 'cheer', 'friend', 'heaven', 'loyal', 'pleasure', 'diamond', 'gentle', 'honest', 'lucky', 'rainbow', 'diploma', 'gift', 'honor', 'miracle', 'sunrise', 'family', 'happy', 'laughter', 'paradise', 'vacation' |
| A22 | Unpleasant | 'abuse', 'crash', 'filth', 'murder', 'sickness', 'accident', 'death', 'grief', 'poison', 'stink', 'assault', 'disaster', 'hatred', 'pollute', 'tragedy', 'divorce', 'jail', 'poverty', 'ugly', 'cancer', 'kill', 'rotten', 'vomit', 'agony', 'prison' |
| A23 | Pleasant 5 | 'caress', 'freedom', 'health', 'love', 'peace', 'cheer', 'friend', 'heaven', 'loyal', 'pleasure', 'diamond', 'gentle', 'honest', 'lucky', 'rainbow', 'diploma', 'gift', 'honor', 'miracle', 'sunrise', 'family', 'happy', 'laughter', 'paradise', 'vacation' |
| A24 | Unpleasant 5 | 'abuse', 'crash', 'filth', 'murder', 'sickness', 'accident', 'death', 'grief', 'poison', 'stink', 'assault', 'disaster', 'hatred', 'pollute', 'tragedy', 'divorce', 'jail', 'poverty', 'ugly', 'cancer', 'kill', 'rotten', 'vomit', 'agony', 'prison' |
| A25 | Pleasant 9 | 'joy', 'love', 'peace', 'wonderful', 'pleasure', 'friend', 'laughter', 'happy' |
| A26 | Unpleasant 9 | 'agony', 'terrible', 'horrible', 'nasty', 'evil', 'war', 'awful', 'failure' |

### A.3 Targets-Attributes Relationship

| Id | Target 1 | Target 2 | Attribute 1 | Attribute 2 | Bias Type |
|---|---|---|---|---|---|
| Q1 | T1 | T4 | A1 | A2 | Gender |
| Q2 | T1 | T4 | A5 | A4 | Gender |
| Q3 | T1 | T4 | A3 | A4 | Gender |
| Q4 | T1 | T4 | A6 | A7 | Gender |
| Q5 | T1 | T4 | A6 | A8 | Gender |
| Q6 | T1 | T4 | A21 | A22 | Gender |
| Q7 | T1 | T4 | A19 | A20 | Gender |
| Q8 | T1 | T4 | A9 | A10 | Gender |
| Q9 | T1 | T4 | A11 | A12 | Gender |
| Q10 | T1 | T4 | A13 | A14 | Gender |
| Q11 | T2 | T5 | A1 | A2 | Gender |
| Q12 | T2 | T5 | A5 | A4 | Gender |
| Q13 | T2 | T5 | A3 | A4 | Gender |
| Q14 | T2 | T5 | A6 | A7 | Gender |
| Q15 | T2 | T5 | A6 | A8 | Gender |
| Q16 | T2 | T5 | A21 | A22 | Gender |
| Q17 | T2 | T5 | A19 | A20 | Gender |
| Q18 | T2 | T5 | A9 | A10 | Gender |
| Q19 | T2 | T5 | A11 | A12 | Gender |
| Q20 | T2 | T5 | A13 | A14 | Gender |
| Q21 | T3 | T6 | A1 | A2 | Gender |
| Q22 | T3 | T6 | A5 | A4 | Gender |
| Q23 | T3 | T6 | A3 | A4 | Gender |
| Q24 | T3 | T6 | A6 | A7 | Gender |
| Q25 | T3 | T6 | A6 | A8 | Gender |
| Q26 | T3 | T6 | A21 | A22 | Gender |
| Q27 | T3 | T6 | A19 | A20 | Gender |
| Q28 | T3 | T6 | A9 | A10 | Gender |
| Q29 | T3 | T6 | A11 | A12 | Gender |
| Q30 | T3 | T6 | A13 | A14 | Gender |
| Q31 | T7 | T8 | A23 | A24 | Ethnicity |
| Q32 | T7 | T11 | A23 | A24 | Ethnicity |
| Q33 | T7 | T9 | A23 | A24 | Ethnicity |
| Q34 | T7 | T10 | A23 | A24 | Ethnicity |

| | | | | | |
|---|---|---|---|---|---|
| Q35 | T7 | T8 | A25 | A26 | Ethnicity |
| Q36 | T7 | T11 | A25 | A26 | Ethnicity |
| Q37 | T7 | T9 | A25 | A26 | Ethnicity |
| Q38 | T7 | T10 | A25 | A26 | Ethnicity |
| Q39 | T7 | T8 | A18 | A15 | Ethnicity |
| Q40 | T7 | T11 | A18 | A16 | Ethnicity |
| Q41 | T7 | T9 | A18 | A17 | Ethnicity |
| Q42 | T7 | T8 | A19 | A20 | Ethnicity |
| Q43 | T7 | T11 | A19 | A20 | Ethnicity |
| Q44 | T7 | T9 | A19 | A20 | Ethnicity |
| Q45 | T7 | T10 | A19 | A20 | Ethnicity |
| Q46 | T12 | T13 | A23 | A24 | Religion |
| Q47 | T12 | T14 | A23 | A24 | Religion |
| Q48 | T13 | T14 | A23 | A24 | Religion |
| Q49 | T12 | T13 | A25 | A26 | Religion |
| Q50 | T12 | T14 | A25 | A26 | Religion |
| Q51 | T13 | T14 | A25 | A26 | Religion |
| Q52 | T12 | T13 | A19 | A20 | Religion |
| Q53 | T12 | T15 | A19 | A20 | Religion |
| Q54 | T13 | T15 | A19 | A20 | Religion |