Gender Bias in Meta-Embeddings

Masahiro Kaneko[1] **Danushka Bollegala**[2,3*] **Naoaki Okazaki**[1]
[1]Tokyo Institute of Technology [2]University of Liverpool [3]Amazon
masahiro.kaneko@nlp.c.titech.ac.jp
danushka@liverpool.ac.uk okazaki@c.titech.ac.jp

Abstract

Different methods have been proposed to develop meta-embeddings from a given set of source embeddings. However, the source embeddings can contain unfair gender-related biases, and how these influence the meta-embeddings has not been studied yet. We study the gender bias in meta-embeddings created under three different settings: (1) meta-embedding multiple sources without performing any debiasing (Multi-Source No-Debiasing), (2) meta-embedding multiple sources debiased by a single method (Multi-Source Single-Debiasing), and (3) meta-embedding a single source debiased by different methods (Single-Source Multi-Debiasing). Our experimental results show that meta-embedding amplifies the gender biases compared to input source embeddings. We find that debiasing not only the sources but also their meta-embedding is needed to mitigate those biases. Moreover, we propose a novel debiasing method based on meta-embedding learning where we use *multiple* debiasing methods on a *single* source embedding and then create a single unbiased meta-embedding.

1 Introduction

Various pre-trained word embeddings have been successfully used as features for representing input texts in many NLP tasks (Dhillon et al., 2015; Mnih and Hinton, 2009; Collobert et al., 2011; Huang et al., 2012; Mikolov et al., 2013a; Pennington et al., 2014a). Combining multiple word embeddings leads to more accurate and exhaustive meta-embeddings in terms of vocabulary, learned expressions etc (Yin and Schütze, 2016). For example, there are meta-embedding methods that use the average of multiple embeddings (Coates and Bollegala, 2018), concatenate multiple embeddings (Bollegala, 2022), use locally-linear (Bollegala et al., 2018) or global (Yin and Schütze, 2016) projections, or use autoencoders (Bao and Bollegala, 2018).

However, the source embeddings can contain unfair gender-related biases (Barrett et al., 2019; Xie et al., 2017; Elazar and Goldberg, 2018; Li et al., 2018). To address these drawbacks, various debiasing methods have been proposed in the literature. For example, many projection-based methods have been proposed to eliminate biases in static word embeddings (Zhao et al., 2018; Kaneko and Bollegala, 2019; Wang et al., 2020). Bolukbasi et al. (2016) proposed a hard-debiasing (**HARD**) method that projects gender-neutral words into a subspace, which is orthogonal to the gender dimension defined by a list of gender-definitional words. Ravfogel et al. (2020) proposed iterative Null-space Projection (**INLP**) debiasing. They found that iteratively projecting word embeddings to the null space of the gender direction improves the debiasing performance. Kaneko and Bollegala (2021b) proposed dict-debiasing (**DICT**) – a method for removing biases from pre-trained word embeddings using dictionaries, without requiring access to the original training resources or any knowledge regarding the word embedding algorithms used.

On the other hand, to the best of our knowledge, the effect on gender bias due to meta-embedding that uses multiple sources has not been investigated. Even if we had perfectly debiased the individual source embeddings, some meta-embedding methods such as averaging do *not* guarantee debiased meta-embeddings as we prove in §4. In this study, we classify meta-embeddings into the following three types from the viewpoint of debiasing and analyze them: (1) **Multi-Source No-Debiasing**: meta-embeddings created from multiple source embeddings without any debiasing; (2) **Multi-Source Single-Debiasing**: meta-embeddings cre-

[*] Danushka Bollegala holds concurrent appointments as a Professor at University of Liverpool and as an Amazon Scholar. This paper describes work performed at the University of Liverpool and is not associated with Amazon.

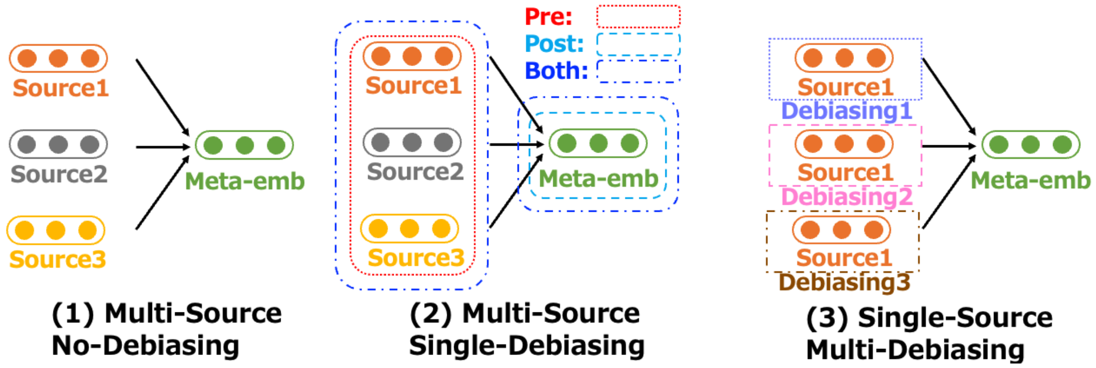

Figure 1: We investigate the bias in three types of meta-embeddings: Multi-Source No-Debiasing, Multi-Source Single-Debiasing and Single-Source Multi-Debiasing. Src denotes the source embeddings and the boxes represent target of debiasing. pre, post, and both indicate as to what stage the debiasing is performed.

ated from multiple source embeddings debiased by a single debiasing method; (3) **Single-Source Multi-Debiasing**: meta-embedding from the same source embedding, debiased using different debiasing methods.[1] Multi-Source Single-Debiasings were examined by debiasing: each source embeddings (**pre**), the learned meta-embeddings (**post**), and both source embeddings and meta-embeddings (**both**). Figure 1 shows how a meta-embedding is learned for those methods.

These methods are agnostic to types of meta-embedding learning algorithms and demonstrate different aspects of debiasing effects. We use Multi-Source No-Debiasings to investigate bias in meta-embeddings learned using existing approaches. The purpose of Multi-Source Single-Debiasing is to investigate how to effectively debias existing meta-embeddings. In Single-Source Multi-Debiasing, we combine the same embeddings debiased by different methods to investigate whether debiasing methods can complement each other's strengths and weaknesses to obtain more effective debiased embeddings. To the best of our knowledge, no studies have been proposed that combine multiple debiasing methods.

We use three debiasing methods and five meta-embeddings in our study. We focus on gender bias, since there are several methods (Bolukbasi et al., 2016; Ravfogel et al., 2020; Kaneko and Bollegala, 2021b) that can be used to combine debiasing methods and datasets (Caliskan et al., 2017; Zhao et al., 2018; Du et al., 2019) that can be examined in different ways. Experimental results show that the gender bias is amplified by meta-embedding methods without any treatment for debiasing. Moreover, the gender bias increases with the number of source embeddings used in the meta-embedding.

Interestingly, we can successfully debias meta-embeddings without losing their superiority of the performance improvements in two out of three word embedding benchmarks. The Multi-Source Single-Debiasing results indicate that debiasing both source embeddings and meta-embeddings is the best practice in two out of three bias evaluation benchmarks. We also demonstrate that Single-Source Multi-Debiasing performs better than using only one debiasing method in all three bias evaluation benchmarks. It can be seen as a debiasing method that uses an ensemble of existing debiasing methods via a meta-embedding framework to create more reliable unbiased embeddings than if we had used a single debiasing method. This is an important result given that there exists a broad range of debiasing methods proposed in the NLP community based on different principles and complementary strengths, yet no single best method exist (Meade et al., 2022; Czarnowska et al., 2021).

2 Meta-Embedding Learning Methods

Depending on whether debiasing methods are applied on source embeddings or their meta-embedding, three variants can be identified: Multi-Source No-Debiasing, Multi-Source Single-Debiasing and Single-Source Multi-Debiasing. To explain these settings further, let us consider a set of N source word embeddings $s_1, s_2, \ldots, s_N$ re-

[1] It is also possible to adapt multiple debiasing methods to multiple source embeddings and learn meta-embedding from them, but this is not the focus of this study because it would increase the vector size (total dimensionality of source embeddings × number of debiasing methods) and computation cost tremendously. For example, in this case, there are four 300-dimensional word embeddings and three debiasing methods, so (4 × 300) × 3 results in a 3600-dimensional vector.

spectively covering vocabularies (i.e. sets of words) $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_N$. The embedding of a word w in s_j is denoted by $\boldsymbol{s}_j(w) \in \mathbb{R}^{d_j}$, where d_j is the dimensionality of $\boldsymbol{s}_j$. In Multi-Source No-Debiasing, meta-embedding $\boldsymbol{m}^{\text{MSND}}(w)$ for word w is computed from $\boldsymbol{s}_1(w), \boldsymbol{s}_2(w), \ldots, \boldsymbol{s}_N(w)$ using some meta-embedding learning method, where MSND represents Multi-Source No-Debiasing. Multi-Source Single-Debiasing and Single-Source Multi-Debiasing are obtained by applying debiasing methods described in § 3. Let us denote debiasing for word embedding $\boldsymbol{s}_j(w)$ as $\boldsymbol{d}(\boldsymbol{s}_j(w))$. In Multi-Source Single-Debiasing, there are three types of debiasing possibilities: *pre*, *post*, and *both*. In *pre*, the debiased source embeddings $\boldsymbol{d}(\boldsymbol{s}_1(w)), \boldsymbol{d}(\boldsymbol{s}_2(w)), \ldots, \boldsymbol{d}(\boldsymbol{s}_N(w))$ are used to computed $\boldsymbol{m}^{\text{MSSDpre}}(w)$, where MSSD represents Multi-Source Single-Debiasing. In *post*, debiasing is performed on the learned meta-embeddings as in $\boldsymbol{m}^{\text{MSSDpost}}(w) = \boldsymbol{d}(\boldsymbol{m}^{\text{MSND}}(w))$. In *both*, debiasing is performed for both pre and post, as in $\boldsymbol{m}^{\text{MSSDboth}}(w) = \boldsymbol{d}(\boldsymbol{m}^{\text{MSSDpre}}(w))$. In Single-Source Multi-Debiasing, we use different debiasing methods for the same source embedding as in $\boldsymbol{d}_1(\boldsymbol{s}_j(w)), \boldsymbol{d}_2(\boldsymbol{s}_j(w)), \ldots, \boldsymbol{d}_M(\boldsymbol{s}_j(w))$ to learn meta-embedding $\boldsymbol{m}^{\text{SSMD}}(w)$. Here, M is the number of debiasing methods and SSMD represents Single-Source Multi-Debiasing.

The source word embeddings, in general, do not have to cover the same set of words. Much prior work in meta-embedding learning assume a common vocabulary over all source embeddings for simplicity. If a particular word is not covered by a source embedding, it can be assigned a zero vector, a randomly initialised vector or we could learn a regression model to predict the missing source embeddings (Yin and Schütze, 2016). Without loss of generality, we will assume that all words for evaluation are covered by a meta-embedding vocabulary $\mathcal{V}$, which is composed by each source embedding's vocabulary $\mathcal{V}_j$, after applying any one of the above-mentioned methods. Here j represents the j-th source embedding's vocabulary. Each word w is assumed to be included in at least one of the vocabularies $\mathcal{V}_j$, and zero embeddings are assigned for $w \notin \mathcal{V}_j$.

We consider five previously proposed meta-embedding learning methods for static word embeddings in this study to learn $\boldsymbol{m}(w)$ for Multi-Source No-Debiasing, Multi-Source Single-Debiasing and Single-Source Multi-Debiasing as follows: concatenation (**CONC** Bollegala, 2022), averaging (**AVG** Coates and Bollegala, 2018), globally-linear meta-embedding (**GLE** Yin and Schütze, 2016), locally-linear meta-embedding (**LLE** Bollegala et al., 2018), and averaged autoencoded meta-embeddings (**AEME** Bao and Bollegala, 2018). According to Bollegala and O'Neill (2022), these are the most widely-used meta-embedding learning methods. They methods are described in detail in Appendix §1.

3 Debiasing Methods for Static Word Embeddings

Different methods have been proposed in prior work for debiasing static word embeddings. We consider the following three popular debiasing methods in this study: (1) hard-debiasing (**HARD** Bolukbasi et al., 2016), (2) Iterative Null Space Projection (**INLP** Ravfogel et al., 2020), and (3) dictionary-based debiasing (**DICT** Kaneko and Bollegala, 2021b). Due to space constraints, we describe those methods in detail in Appendix §2. By adapting these debiasing methods to the meta-embeddings described in §2, we investigate Multi-Source Single-Debiasing and Single-Source Multi-Debiasing.

4 AVG does *not* Protect HARD Debiasing

In general, it is difficult to mathematically analyze the gender bias in debiased source embeddings when they are meta-embedded using a particular meta-embedding learning method. However, such an analysis is possible in the special case of the HARD debiasing for CONC and AVG, and shows that even if all source embeddings are debiased their meta-embedding might not always remain debiased.

Let us consider applying HARD to debias two sources s_1, s_2 independently and create their meta-embeddings separately using CONC and AVG. To simplify the discussion, let us assume that both s_1 and s_2 to be k-dimensional and having bias vector sets respectively $\{\boldsymbol{b}_1^{(1)}, \ldots, \boldsymbol{b}_k^{(1)}\}$ and $\{\boldsymbol{b}_1^{(2)}, \ldots, \boldsymbol{b}_k^{(2)}\}$. We debias the source embeddings of a word w using HARD and obtain $\boldsymbol{d}_1(w)$ and $\boldsymbol{d}_2(w)$, given respectively by (1) and (2).

$$\boldsymbol{d}_1(w) = \frac{\boldsymbol{s}_1(w) - \boldsymbol{w}_{1,\mathcal{B}}}{||\boldsymbol{s}_1(w) - \boldsymbol{w}_{1,\mathcal{B}}||} \tag{1}$$

$$\boldsymbol{d}_2(w) = \frac{\boldsymbol{s}_2(w) - \boldsymbol{w}_{2,\mathcal{B}}}{||\boldsymbol{s}_2(w) - \boldsymbol{w}_{2,\mathcal{B}}||} \tag{2}$$

Here, $\boldsymbol{w}_{1,\mathcal{B}}$ and $\boldsymbol{w}_{2,\mathcal{B}}$ denote the projected source embeddings of w onto the gender subspaces in each source embedding spaces s_1 and s_2. Let us denote the concatenated meta-embedding of $\boldsymbol{d}_1(w)$ and $\boldsymbol{d}_2(w)$ by $\boldsymbol{m}_{\text{conc}}(w)$. Consider the bias vector $\boldsymbol{b}_j^{(1)} \oplus \boldsymbol{b}_j^{(2)}$. Because the inner-product decomposes over the individual components under vector concatenation we can simplify $\langle \boldsymbol{m}_{\text{conc}}(w), \boldsymbol{b}_j^{(1)} \oplus \boldsymbol{b}_j^{(2)} \rangle$ as follows:

$$\langle \boldsymbol{d}_1(w), \boldsymbol{b}_j^{(1)} \rangle + \langle \boldsymbol{d}_2(w), \boldsymbol{b}_j^{(2)} \rangle \tag{3}$$

Each term in (3) are separately zero because the debiased embeddings are orthogonal to the bias vectors by construction in each source. Therefore, concatenated meta-embedding preserves the debiasing result under HARD debiasing.

However, this is not true for other meta-embedding methods such as averaging. To see this, consider $\langle \boldsymbol{d}_1(w) + \boldsymbol{d}_2(w), \boldsymbol{b}_j^{(1)} + \boldsymbol{b}_j^{(2)} \rangle$, which results in four terms as in (4).

$$\begin{aligned} &\langle \boldsymbol{d}_1(w), \boldsymbol{b}_j^{(1)} \rangle + \langle \boldsymbol{d}_2(w), \boldsymbol{b}_j^{(2)} \rangle \\ &+ \langle \boldsymbol{d}_1(w), \boldsymbol{b}_j^{(2)} \rangle + \langle \boldsymbol{d}_2(w), \boldsymbol{b}_j^{(1)} \rangle \end{aligned} \tag{4}$$

Note that the first two terms in (4) are zero because they are in the same vector space and the inner-products are taken w.r.t. to the corresponding bias vectors. However, the last two terms in (4) are *not* generally zero. Therefore, AVG does not generally preserve the HARD debiasing result.

5 Experiments

Our goal in this paper is to evaluate whether gender bias is amplified and to what degree by the different meta-embedding learning methods. However, this bias amplification must be considered relative to the accuracy of the semantic representations produced by those meta-embedding learning methods. For example, a meta-embedding learning method can produce perfectly unbiased representations by mapping all words to a constant vector, which is useless for any downstream task requiring semantic representations of words. For this reason, we conduct our evaluations using two types of datasets to evaluate gender biases in the debiased embeddings, while preserving useful semantic information necessary for downstream tasks: (a) bias evaluation datasets covering different types of gender biases (described in §5.1) and (b) word embedding benchmarks related to semantic similarity prediction and POS tagging tasks (described in §5.2).

5.1 Bias Evaluation Benchmarks

We use Word Embedding Association Test (**WEAT**; Caliskan et al., 2017), Word Association Test (**WAT**; Du et al., 2019) and SemBias (**SB**; Zhao et al., 2018) for bias evaluation. The closer the scores of all of these evaluations to 0, the less bias there is.

WEAT: WEAT (Caliskan et al., 2017) quantifies various biases (e.g., gender, race, and age) using semantic similarities between word embeddings. It compares two same size sets of *target* words $\mathcal{X}$ and $\mathcal{Y}$ (e.g. European and African names), with two sets of *attribute* words $\mathcal{A}$ and $\mathcal{B}$ (e.g. *pleasant* vs. *unpleasant*). The bias score, $s(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B})$, for each target is calculated as follows:

$$\begin{aligned} s(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}) = &\sum_{\boldsymbol{x} \in \mathcal{X}} k(\boldsymbol{x}, \mathcal{A}, \mathcal{B}) \\ &- \sum_{\boldsymbol{y} \in \mathcal{Y}} k(\boldsymbol{y}, \mathcal{A}, \mathcal{B}) \end{aligned} \tag{5}$$

$$\begin{aligned} k(\boldsymbol{t}, \mathcal{A}, \mathcal{B}) = &\ \text{mean}_{\boldsymbol{a} \in \mathcal{A}} f(\boldsymbol{t}, \boldsymbol{a}) \\ &- \text{mean}_{\boldsymbol{b} \in \mathcal{B}} f(\boldsymbol{t}, \boldsymbol{b}) \end{aligned} \tag{6}$$

Here, f is the cosine similarity between the word embeddings. The one-sided p-value for the permutation test regarding $\mathcal{X}$ and $\mathcal{Y}$ is calculated as the probability of $s(\mathcal{X}_i, \mathcal{Y}_i, \mathcal{A}, \mathcal{B}) > s(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B})$. The effect size is calculated as the normalised measure given by (7).

$$\frac{\text{mean}_{x \in \mathcal{X}} s(x, \mathcal{A}, \mathcal{B}) - \text{mean}_{y \in \mathcal{Y}} s(y, \mathcal{A}, \mathcal{B})}{\text{sd}_{t \in \mathcal{X} \cup \mathcal{Y}} s(t, \mathcal{A}, \mathcal{B})} \tag{7}$$

WEAT can evaluate eight types of bias, and we report the average absolute effect sizes for T4, T5 and T6, which are related to gender bias.

WAT: WAT[2] is a method to measure gender bias over a large set of words (Du et al., 2019). It calculates the gender information vector for each word in a word association graph created with Small World of Words project (SWOWEN; Deyne et al., 2019) by propagating information related to masculine and feminine gender pair set (e.g. *she* and *he*) $(w_m^i, w_f^i) \in \mathcal{L}$, using a random walk approach (Zhou et al., 2003). The gender information

[2] https://github.com/Yupei-Du/bias-in-wat

is represented as a 2-dimensional vector (b_m, b_f), where b_m and b_f denote respectively the masculine and feminine orientations of a word. The gender information vectors of masculine words, feminine words, and other words are initialised respectively with vectors (1, 0), (0, 1), and (0, 0). The bias score of a word is defined as $\log(b_m/b_f)$. We evaluate the gender bias of word embeddings using the Pearson correlation coefficient between the bias score of each word and the score given by (8) computed as the averaged difference of cosine similarities between masculine and feminine words.

$$\frac{1}{|\mathcal{L}|}\sum_{i=1}^{|\mathcal{L}|}\left(f(w, w_m^i) - f(w, w_f^i)\right) \qquad (8)$$

SB: The SB dataset (Zhao et al., 2018)[3] contains three categories of word-pairs: (1) **Definition**, a gender-definition word pair (e.g. hero – heroine), (2) **Stereotype**, a gender-stereotype word pair (e.g., manager – secretary) and (3) **None**, two other word-pairs with similar meanings unrelated to gender (e.g., jazz – blues, pencil – pen). The SB metric uses the cosine similarity between the $\overrightarrow{he} - \overrightarrow{she}$ gender directional vector and $\boldsymbol{a} - \boldsymbol{b}$ for each word pair (a – b) in the above categories to measure gender bias. SB contains 20 Stereotype word pairs and 22 Definition word pairs and uses the Cartesian product to generate 440 instances. Zhao et al. (2018) used a subset of 40 instances associated with two seed word-pairs, not used in the word list for training, to evaluate the generalisability of a debiasing method. We expect high similarity scores in the Definition category and low similarity scores in the Stereotype and None categories for unbiased word embeddings. This paper reports the percentage of times that pairs of Stereotype and None categories had the highest similarity in the subset and this score is expected to be low.

5.2 Word Embedding Benchmarks

We use SimLex (**SL**; Hill et al., 2015), **MEN** (Bruni et al., 2012) and CoNLL-2003 POS tagging (**POS tagging**; Zhao et al., 2018) as word embedding benchmarks. The higher these scores, the better the performance.

	WEAT	WAT	SB	SL	MEN	POS
W2V	1.31	0.47	17.0	44.2	78.2	87.8
GV	1.17	**0.58**	17.0	40.8	80.5	90.9
FTC	1.31	0.53	13.2	**47.1**	81.5	88.7
FTW	1.08	0.50	15.2	44.1	80.1	80.1
ALL	1.22	0.52	15.6	44.1	80.1	88.8
AVG	**1.46**	0.53	18.0	41.7	80.5	89.7
CONC	1.33	**0.58**	16.6	42.7	81.3	**91.2**
LLE	1.39	0.56	**30.0**	44.2	80.8	89.0
GLE	1.31	0.52	16.8	43.7	**82.1**	87.7
AEME	1.28	0.53	11.1	43.7	81.1	89.1

Table 1: The results of bias evaluation and word embedding benchmarks for source embeddings (W2V, GV, FTC and FTW) and meta-embeddings of Multi-Source No-Debiasing (AVG, CONC, LLE, GLE and AEME). ALL is the score to compare with the score of Multi-Source No-Debiasing, which is arithmetic mean over the score of source embeddings. **Bold** indicates the results with the highest bias and performance.

Semantic Similarity: The semantic similarity between two words is calculated as the cosine similarity between their word embeddings and compared against the human ratings using the Spearman correlation coefficient. Following prior work, we use SL (Hill et al., 2015) and MEN (Bruni et al., 2012) for evaluations.

POS tagging: To evaluate the performance in a downstream task that uses word/meta embeddings as input representations, we evaluate the performance of POS tagging of the model initialised by the pre-trained word embedding. We use the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003) for training and evaluating the POS tagger, implemented as a single LSTM with a 100-dimensional hidden layer. All the weights and biases of LSTM are initialized from $\mathcal{U}(-\sqrt{k}, \sqrt{k})$ where $\mathcal{U}$ is a uniform distribution and $k = \frac{1}{\text{hidden\_size}}$. We optimise the model using SGD with a learning rate of 0.1. We set the batch size to 32 and report results of the model on the test data. The model with the best performance was selected using the development data in 10 epochs. We use WEAT T4 as the development data.

5.3 Settings

In our experiments, we use the following publicly available pre-trained word embeddings as the source embeddings: Word2Vec[4] (**W2V**; Mikolov et al., 2013b) is 300-dimensional embeddings for 3M words trained on Google News corpus,

[3] https://github.com/uclanlp/gn_glove

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

	num	WEAT	WAT	SB	SL	MEN	POS
ALL	1	1.22	0.52	15.6	44.1	80.1	88.8
AVG	2	1.32	0.51	17.3	43.0	79.9	89.1
	3	1.38	**0.53**	17.5	42.8	80.0	89.6
	4	**1.46**	**0.53**	**18.0**	41.7	**80.5**	**89.7**
CONC	2	1.26	0.53	16.4	43.1	80.3	90.3
	3	1.30	0.57	16.5	42.2	80.6	90.8
	4	**1.33**	**0.58**	**16.6**	42.7	**81.3**	**91.2**
LLE	2	1.25	0.52	21.7	43.3	80.3	87.9
	3	1.29	0.54	25.2	43.9	80.5	88.4
	4	**1.39**	**0.56**	**30.0**	**44.2**	**80.8**	**89.0**
GLE	2	1.26	0.50	16.0	43.3	81.0	87.6
	3	1.29	0.50	16.3	43.5	81.3	87.6
	4	**1.31**	**0.52**	**16.8**	43.7	**82.1**	87.7
AEME	2	1.24	0.52	12.0	43.3	80.5	88.9
	3	1.25	0.52	12.6	43.5	80.8	89.0
	4	**1.28**	**0.53**	11.1	43.7	**81.1**	**89.1**

Table 2: The results of bias evaluation and word embedding of benchmarks of Multi-Source No-Debiasing with AVG, CONC, LLE, GLE and AEME using different number of source embeddings. This is the result of arithmetic mean scores for each number. Here, num=1 represents the arithmetic mean of the results of all source embeddings. **Bold** indicates the results with the highest bias and performance in num=2, 3, 4, considering num=1, 2, 3, 4.

Method		WEAT	WAT	SB	SL	MEN	POS
ALL		1.22	0.52	15.6	44.1	80.1	88.8
HARD		0.93	0.45	7.7	44.2	80.0	88.5
INLP		0.91	0.43	12.2	43.6	79.2	88.7
DICT		0.97	0.51	12.9	**47.2**	**82.1**	**88.8**
HARD	pre	0.93	0.48	9.7	43.2	79.5	87.1
	post	0.84	0.50	9.2	42.4	**79.9**	**87.9**
	both	0.86	0.40	9.1	**44.1**	79.7	87.7
INLP	pre	0.95	0.49	13.2	42.2	**79.3**	88.2
	post	0.91	0.48	12.5	41.1	**79.3**	**88.7**
	both	0.86	0.37	12.2	**44.2**	79.1	88.6
DICT	pre	1.01	0.50	14.2	46.1	**84.9**	89.0
	post	0.97	0.51	13.8	45.3	85.2	**90.1**
	both	0.89	0.44	14.1	**46.3**	84.6	89.9

Table 3: The results of bias evaluation and word embeddings benchmarks of source embeddings (ALL), debiased source embeddings (HARD, INLP, DICT) and meta-embeddings of Multi-Source Single-Debiasing (pre, post, both). Here, pre indicates debiasing source embeddings then learning meta-embeddings, post indicates debiasing the meta-embeddings, and both indicates debiasing both source and meta-embeddings. pre, post and both average the scores using the five meta-embedding methods. ALL, HARD, INLP and DICT scores are the average results of W2V, GV, FTC and FTW. Underline represents the most debiased results and **Bold** shows the highest performance.

GloVe[5] (**GV**; Pennington et al., 2014b) is 300-dimensional embeddings for 2.2M words trained on the Common Crawl and FastText[6] (**FTC** and **FCW**; Bojanowski et al., 2017) are 300-dimensional embeddings for 2M words trained on Common Crawl and Wikipedia.

We used the publicly available code by the original authors for **HARD**[7], **INLP**[8] and **DICT**[9] debiasing methods with the default hyperparameters and word lists for training used in the original implementations. Debiasing requires less than half an hour in all experiments on a GeForce RTX 2080 Ti GPU.

5.4 Gender Bias in Meta-Embeddings

To study how different Multi-Source No-Debiasing methods amplify the gender bias in the source embeddings, in Table 1 we compare source and meta-embeddings using the datasets described in §5. From Table 1 we see that different source embeddings express different levels of gender biases. Among the source embeddings, we see that GV has the highest bias in WEAT, WAT and SB. On the other hand, FTC has the best performance in SL and MEN, whereas GV has the best performance in POS. Here, **ALL** is the arithmetic mean of the scores for the four source embeddings W2V, GV, FTC and FTW, which simulates the setting where sources are used separately without creating any meta-embeddings. We use ALL to compare with the results of Multi-Source No-Debiasing.

Among the Multi-Source No-Debiasing methods, we see that LLE has a lower bias on WEAT and WAT, but its performance is considerably worse in MEN and POS. Ideally, we would prefer Multi-Source No-Debiasing methods that combine the information in multiple sources, while not aggregating or amplifying any gender bias present in the source embeddings. In this regard, we can see that AEME, which uses autoencoders to learn meta-embeddings, has a lower bias as well as better performance. This result aligns well with prior proposals where Kaneko and Bollegala (2019) used autoencoders to debias static word embeddings.

[5] https://github.com/stanfordnlp/GloVe

[6] https://fasttext.cc/docs/en/english-vectors.html

[7] https://github.com/tolga-b/debiaswe

[8] https://github.com/shauli-ravfogel/nullspace_projection

[9] https://github.com/kanekomasahiro/dict-debias

Moreover, it has been shown that autoencoding improves pre-trained word embeddings by making them more isotropic (Kaneko and Bollegala, 2020), which might explain the superior performance of AEME as a meta-embedding learning method.

Considering that Multi-Source No-Debiasing methods use multiple source embeddings as the input, an interesting open question is whether more sources result in more biased meta-embeddings. To study this relationship empirically, in Table 2 we use varying numbers of source embeddings and create meta-embeddings. For example, the row corresponding to num=2 in Table 2 shows the setting where we use two out of the available four source embeddings to create meta-embeddings. This results in six (${}_4C_2$) different meta-embeddings produced in num=2 setting. We evaluate each of those meta-embeddings for the bias and semantic representation ability and report the arithmetic mean. Note that num=1 setting corresponds to the previously described **ALL** baseline, which reports the average scores when each source embedding is evaluated individually, without creating their meta-embeddings. num=4 are the same as AVG, CONC, LLE, GLE, AEME in Table 1.

From Table 2 we see that, except for AEME in SB, the gender bias is amplified when more sources are used in the meta-embedding process. Moreover, the average performances of meta-embeddings increase with the number of sources. In Multi-Source No-Debiasing, we can see that increasing the number of source embeddings amplifies the bias as well as improving the task performance.

5.5 Debiasing vs. Meta-Embedding

Next, we study the effectiveness of Multi-Source Single-Debiasing using different debiasing methods described in § 3 for removing unfair gender bias from Multi-Source No-Debiasing. Note that debiasing and meta-embedding learning methods can be applied to a given set of source embeddings in an arbitrary order. Here we consider three settings: **pre** (first debias the sources and then create their meta-embedding), **post** (first create meta-embedding of the sources and then debias it), and **both** (apply debiasing to the source as well as meta-embeddings). For **pre**, **post** and **both** of Multi-Source Single-Debiasing, we use five meta-embedding learning methods and reported their arithmetic mean scores.[10] Table 3 shows the arithmetic mean results of applying HARD, INLP, and DICT-debiasing compared to all the source embeddings and the results of Multi-Source Single-Debiasing. First, we see that all debiasing methods reduce the gender bias in the source embeddings compared to ALL. In particular, DICT-debiasing not only debiases the source embeddings but also improves their performance, as can be seen from the evaluations on SL and MEN datasets.

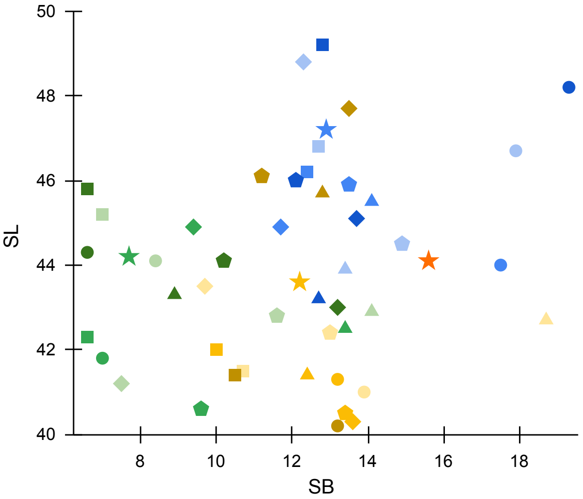

Figure 2: Scores for SB and SL. SB score is horizontal axis and SL score is vertical axis.

We see that HARD, INLP and DICT debiasing methods for a single source are still effective at debiasing meta-embeddings created from multiple sources. Furthermore, **both** outperforms **pre** and **post** for debiasing meta-embeddings except HARD both on WEAT, while not degrading performance on SL, MEN and POS. Although we do not show the individual results of meta-embedding in Table 3, out of 45 results (i.e. 3 debiasing methods × 5 meta-embeddings × 3 test data) there are 5 cases where **both** under-performs in bias evaluation to **pre** or **post**. Moreover, the tendency for **both** to obtain the best debiasing results does not depend on the meta-embedding method used.

Because scores in Table 3 are averaged over meta-embedding methods, to further analyse effects due to each meta-embedding method, we plot all combinations in Figure 2, where x-axis shows SB scores (lower values indicate debiased embeddings) and y-axis shows SL scores (higher values indicate accurate embeddings). The shapes of the

[10] If the results for each meta-embedding method are listed in Table 3, there will be 5 rows for each of pre, post and both, and 9 (number of pre, post and both) × 5 (number of meta-embedding) + 5 (rows other than pre, post and both) will make a huge table of 50 rows, so for the reason of space, the meta-embedding scores are averaged.

	Emb.	WEAT	WAT	SB	SL	MEN	POS
Source	W2V	1.31	0.47	17.0	44.2	78.2	87.8
	GV	1.17	0.58	17.0	**40.8**	80.5	90.9
	FTC	1.31	0.53	13.2	**47.1**	81.5	88.7
	FTW	1.08	0.50	15.2	**44.1**	80.1	80.1
Debiased	W2V	1.08	0.46	11.7	44.2	**80.8**	**90.8**
	GV	1.01	0.52	11.9	40.5	**81.5**	90.7
	FTC	1.14	0.51	12.5	46.8	78.9	89.4
	FTW	0.94	0.47	13.4	44.0	82.0	**81.5**
SSMD	W2V	0.98	0.44	9.8	**44.3**	80.7	90.5
	GV	0.90	0.32	11.0	40.5	**81.5**	**91.1**
	FTC	1.05	0.41	8.9	45.3	**81.8**	**89.8**
	FTW	0.78	0.44	12.1	43.0	**82.3**	81.0

Table 4: The results of bias evaluation and word embedding benchmarks of source, debiased source and Single-Source Multi-Debiasing embeddings (SSMD). The results of each meta-embedding learning method are arithmetic means in SSMD. Underline represents the most debiased results per embedding method.

points denote the meta-embedding methods: source - star; AVG - circle; CONC - square; LLE - triangle; GLE - pentagon; and AEME - diamond, respectively. The colors indicate the debiasing methods: No-Debiasing - orange; HARD - green; INLP - yellow; DICT - blue, respectively. The density of the colours indicates pre: light, post: intermediate, and both: dark, respectively.

Except for some results of DICT, meta-embeddings are debiased regardless of the debiasing method used compared to their source embeddings. In most cases, CONC results in the most debiased meta-embeddings compared to the debiased source embeddings. In addition, HARD and DICT of CONC show improved SL performance compared to their source embeddings. The performance of CONC is considered to be higher than that of source embeddings because the number of dimensions of CONC is larger and more expressive than that of source embeddings.

5.6 The Same Source Embeddings, Different Debiasing

In § 5.5 we observed that each debiasing method has its own strengths and weaknesses in removing bias-related information and preserving useful semantic information in word embeddings. Motivated by this, we propose Single-Source Multi-Debiasing – given a pre-trained source embedding, we first apply different debiasing methods to create multiple debiased versions of that source embedding and subsequently meta-embed them. Specifically, we create debiased versions of a source embedding using HARD, INLP, and DICT debiasing methods separately, and then use AVG, CONC, LLE, GLE to create corresponding meta-embeddings.

Table 4 shows the results for the original source embeddings (Source), debiased source embeddings (Debiased source) and Single-Source Multi-Debiasing embeddings. Here, Single-Source Multi-Debiasing shows the arithmetic mean of the scores of the five meta-embedding learning methods.[11] Moreover, debiased source shows the arithmetic mean of the scores of adapting the three debiasing methods separately to compare the methods using multiple debiasing methods. We see that the bias evaluation of all Single-Source Multi-Debiasing is better than source and debiased source. This indicates that Single-Source Multi-Debiasing improves the overall debiasing performance by taking into account the strengths and weaknesses of each debiasing method. Furthermore, the highest number of scores in SL, MEN and POS indicates that Single-Source Multi-Debiasing is able to learn the highest quality embeddings. Although we do not put the individual results of meta-embedding in Table 4, out of 60 results (i.e. 4 source embeddings × 5 meta-embeddings × 3 test data), Single-Source Multi-Debiasing underperforms only in 4 cases compared to the debiased sources.

6 Conclusion

We studied the gender bias due to meta-embedding under three settings: (1) Multi-Source No-Debiasing, (2) Multi-Source Single-Debiasing (3) Single-Source Multi-Debiasing created from static word embeddings as sources. Our experimental results show that although meta-embedding of Multi-Source No-Debiasing improves performance over the input source embeddings, at the same time, it amplifies the unfair gender bias encoded in the source embeddings. Furthermore, the level of gender bias encoded in a meta-embedding increases with the number of source embeddings used. We found that Multi-Source Single-Debiasing using previously proposed debiasing methods for static word embeddings can be effectively used to debiase meta-embeddings as well. Furthermore, we proposed Single-Source Multi-Debiasing that com-

[11] If the results each meta-embedding method are listed in Table 4, there will be 5 rows for each word embedding, and 4 × 5 + 9 will make a huge table of 29 rows, thus due to limited space the scores are averaged.

bines the outputs from multiple debiasing methods and then create a single embedding via a meta-embedding learning method.

7 Limitations

In this paper, we limited our investigation to meta-embedding learning methods applicable to static word embeddings because they are still extensively used in various NLP applications for input representation, particularly in resource/energy constrained devices without GPUs due to their relatively lightweight nature compared to contextualised embeddings obtained from large-scale neural language models (Strubell et al., 2019). However, there has been recent work studying the gender bias in contextualised embeddings (Zhao et al., 2019; Vig, 2019; Bordia and Bowman, 2019; May et al., 2019; Kaneko and Bollegala, 2021a,c; Kaneko et al., 2022; Zhou et al., 2022; Schick et al., 2021). On the other hand, learning meta-embeddings of contextualised embeddings is relatively underdeveloped (Poerner et al., 2020). Therefore, we defer the study of gender bias in contextualised meta-embeddings to future work. Furthermore, in future, we plan to study other types of social biases such as racial and religious biases in meta-embeddings.

8 Ethical Considerations

The goal of our paper was to study the gender bias in various meta-embeddings created in three different settings. We did not manually annotate novel social bias datasets, proposed novel bias evaluation measures nor debiasing methods. Therefore, we do not see any ethical issues arising due to data annotation, or via proposals of novel evaluation metrics or debiasing methods.

The gender biases we considered in this paper cover only binary gender. The bias evaluation in word embeddings used in our paper can evaluate only binary gender. However, gender biases have been reported related to non-binary gender as well (Cao and Daumé III, 2020; Dev et al., 2021). Studying the non-binary gender for debiasing meta-embeddings is an essential next step.

This paper does not cover all debiasing methods for word embeddings (Kaneko and Bollegala, 2019; Wang et al., 2020) and does not guarantee results with any given debiasing method. Furthermore, it should be noted that there may be bias when using debiased meta-embeddings in a downsteream task. It is known that the results of task-independent bias evaluation do not necessarily coincide with the bias evaluation in the downstream task (Goldfarb-Tarrant et al., 2021; Cao et al., 2022).

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proc. of ICLR*.

Cong Bao and Danushka Bollegala. 2018. Learning word meta-embeddings by autoencoding. In *Proc. of COLING*, pages 1650–1661.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. 2019. Adversarial removal of demographic attributes revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6329–6334, Hong Kong, China. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Danushka Bollegala. 2022. Learning meta word embeddings by unsupervised weighted concatenation of source embeddings. In *Proc. of the 31st International Joint Conference on Artificial Intelligence (IJCAI-ECAI)*.

Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. 2018. Think globally, embed locally — locally linear meta-embedding of words. In *Proc. of IJCAI-EACI*, pages 3970–3976.

Danushka Bollegala and James O'Neill. 2022. A survey on word meta-embedding learning. In *Proc. of the 31st International Joint Conference on Artificial Intelligence (IJCAI-ECAI)*.

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

Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145. Association for Computational Linguistics.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356:183–186.

Yang Trista Cao and Hal Daumé III. 2020. Toward gender-inclusive coreference resolution. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4568–4595, Online. Association for Computational Linguistics.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the Intrinsic and Extrinsic Fairness Evaluation Metrics for Contextualized Language Representations. In *Proc. of ACL*.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proc. of NAACL-HLT*, pages 194–198.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuska. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493 – 2537.

Paula Czarnowska, Yogarshi Vyas, and Kashif Shah. 2021. Quantifying social biases in NLP: A generalization and empirical comparison of extrinsic fairness metrics. *Transactions of the Association for Computational Linguistics*, 9:1249–1267.

Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. Harms of gender exclusivity and challenges in non-binary representation in language technologies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1994, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simon De Deyne, Danielle J. Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. 2019. The “small world of words” english word association norms for over 12,000 cue words. *Behavior Research Methods*, 51:987–1006.

Paramveer S. Dhillon, Dean P. Foster, and Lyle H. Ungar. 2015. Eigenwords: Spectral word embeddings. *Journal of Machine Learning Research*, 16:3035–3078.

Yupei Du, Yuanbin Wu, and Man Lan. 2019. Exploring human gender stereotypes with word association test. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6132–6142, Hong Kong, China. Association for Computational Linguistics.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial Removal of Demographic Attributes from Text Data. In *Proc. of EMNLP*.

Josu Goikoetxea, Eneko Agirre, and Aitor Soroa. 2016. Single or multiple? combining word representations independently learned from text and wordnet. In *Proc. of AAAI*, pages 2608–2614.

Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya, and Adam Lopez. 2021. Intrinsic bias metrics do not correlate with application bias. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1926–1940, Online. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proc. of ACL*, pages 873–882.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1641–1650, Florence, Italy. Association for Computational Linguistics.

Masahiro Kaneko and Danushka Bollegala. 2020. Autoencoding improves pre-trained word embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1699–1713, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Masahiro Kaneko and Danushka Bollegala. 2021a. Debiasing pre-trained contextualised embeddings. In *Proc. of 16th conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Masahiro Kaneko and Danushka Bollegala. 2021b. Dictionary-based debiasing of pre-trained word embeddings. In *Proc. of 16th conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Masahiro Kaneko and Danushka Bollegala. 2021c. Unmasking the mask–evaluating social biases in masked language models. *arXiv preprint arXiv:2104.07496*.

Masahiro Kaneko, Aizhan Imankulova, Danushka Bollegala, and Naoaki Okazaki. 2022. Gender bias in masked language models for multiple languages. In *Proc. of NAACL-HLT*.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30. Association for Computational Linguistics.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1878–1898, Dublin, Ireland. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013a. Efficient estimation of word representation in vector space. In *Proc. of ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

G. Miller and W. Charles. 1998. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Andriy Mnih and Geoffrey E. Hinton. 2009. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Proc. of NIPS*, pages 1081–1088.

Avo Muromägi, Kairit Sirts, and Sven Laur. 2017. Linear ensembles of word embedding models. In *Proc. of the Nordic Conference on Computational Linguistics*, pages 96–104.

Jeffery Pennington, Richard Socher, and Christopher D. Manning. 2014a. Glove: global vectors for word representation. In *Proc. of EMNLP*, pages 1532–1543.

Jeffery Pennington, Richard Socher, and Christopher D. Manning. 2014b. Glove: global vectors for word representation. In *EMNLP*, pages 1532–1543.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. Sentence meta-embeddings for unsupervised semantic textual similarity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7027–7034, Online. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Computing Research Repository*, arXiv:2103.00453.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Jesse Vig. 2019. Visualizing Attention in Transformer-Based Language Representation Models.

Tianlu Wang, Xi Victoria Lin, Nazneen Fatema Rajani, Bryan McCann, Vicente Ordonez, and Caiming Xiong. 2020. Double-hard debias: Tailoring word embeddings for gender bias mitigation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5443–5453, Online. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *Proc. of NIPS*.

Wenpeng Yin and Hinrich Schütze. 2016. Learning meta-embeddings by using ensembles of embedding sets. In *Proc. of ACL*, pages 1351–1360.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 629–634, Minneapolis, Minnesota. Association for Computational Linguistics.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning Gender-Neutral Word Embeddings. In *Proc. of EMNLP*, pages 4847–4853.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *NIPS*.

Yi Zhou, Masahiro Kaneko, and Danushka Bollegala. 2022. Sense embeddings are also biased–evaluating social biases in static and contextualised sense embeddings. In *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics*.

A Meta-Embedding Learning Methods

A.1 Concatenation (CONC)

One of the simplest approaches to create a meta-embedding under the unsupervised setting is vector concatenation (Bao and Bollegala, 2018; Yin and Schütze, 2016; Bollegala et al., 2018). Denoting concatenation by $\oplus$, we can express the concatenated meta-embedding, $\boldsymbol{m}_{\text{conc}}(w) \in \mathbb{R}^{d_1 + \ldots + d_N}$, of a word $w \in \mathcal{V}$ by (9).

$$\begin{aligned} \boldsymbol{m}_{\text{conc}}(w) &= \boldsymbol{s}_1(w) \oplus \ldots \oplus \boldsymbol{s}_N(w) \\ &= \oplus_{j=1}^{N} \boldsymbol{s}_j(w) \end{aligned} \tag{9}$$

Goikoetxea et al. (2016) showed the concatenation of word embeddings learnt separately from a corpus and the WordNet produces superior word embeddings. However, one disadvantage of using concatenation to produce meta-embeddings is that it increases the dimensionality of the meta-embedding space, which is the sum of the dimensionalities of the sources.

A.2 Averaging (AVG)

Source embeddings are trained independently and can have different dimensionalities. Even when the dimensionalities do agree, vectors that lie in different vector spaces cannot be readily averaged. However, rather surprisingly, Coates and Bollegala (2018) showed that accurate meta-embeddings can be produced by first zero-padding source embeddings as necessary to bring them to the same dimensionality and then by averaging them to create $\boldsymbol{m}_{\text{avg}}(w)$ as given by (10) when some orthogonality conditions are satisfied by the embedding spaces.

$$\boldsymbol{m}_{\text{avg}}(w) = \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{s}_j^*(w) \tag{10}$$

Here, $\boldsymbol{s}_j^*(w)$ is the zero-padded version of $\boldsymbol{s}_j(w)$ such that its dimensionality is equal to $\max(d_1, \ldots, d_N)$. In contrast to concatenation, averaging has the desirable property that the dimensionality of the meta-embedding is upper-bounded by $\max(d_1, \ldots, d_N) < \sum_{j=1}^{N} d_j$.

A.3 Linear Projections (GLE and LLE)

In their pioneering work on meta-embedding, Yin and Schütze (2016) proposed to project source embeddings to a common space via source-specific linear transformations, which they refer to as 1TON. They require that the meta-embedding of a word w, $\boldsymbol{m}_{\text{1TON}}(w) \in \mathbb{R}^{d_m}$, to be reconstructed from each source embedding, $\boldsymbol{s}_j(w)$ of w. For that they use a linear projection matrix, $\mathbf{A}_j \in \mathbb{R}^{d_j \times d_m}$, from s_j to the meta-embedding space as given by (11).

$$\hat{\boldsymbol{s}}_j(w) = \mathbf{A}_j \boldsymbol{m}_{\text{1TON}}(w) \tag{11}$$

Here, $\hat{\boldsymbol{s}}_j(w)$ is the reconstructed source embedding of w from the meta-embedding $\boldsymbol{m}_{\text{1TON}}(w)$. Next, the squared Euclidean distance between the source- and meta-embeddings is minimised over all words in the intersection of the source vocabularies, subjected to Frobenius norm regularisation as in (12).

$$\underset{\substack{\forall_{j=1}^{N} \mathbf{A}_j \\ \forall_{w \in \mathcal{V}} \ \boldsymbol{m}_{\text{1TON}}(w)}}{\text{minimise}} \sum_{j=1}^{N} \alpha_j \left(\sum_{w \in \mathcal{V}} ||\hat{\boldsymbol{s}}_j(w) - \boldsymbol{s}_j(w)||_2^2 + ||\mathbf{A}_j||_F^2 \right) \tag{12}$$

They use different weighting coefficients α_j to account for the differences in accuracies of the sources. They determine α_j using the Pearson correlation coefficients computed between the human similarity ratings and cosine similarity computed using the each source embedding between word pairs on the Miller and Charles (1998) dataset. The parameters can be learnt using stochastic gradient descent, alternating between projection matrices and meta-embeddings.

Muromägi et al. (2017) showed that by requiring the projection matrices to be orthogonal (corresponding to the Orthogonal Procrustes Problem), the accuracy of the learnt meta-embeddings is further improved. However, 1TON requires all words to be represented in all sources.

Assuming that a single *global linear* (**GLE**) projection can be learnt between the meta-embedding space and each source embedding as done by Yin and Schütze (2016) having all words in all sources is a more vital requirement. Bollegala et al. (2018) relaxed this requirement by learning *locally linear* (**LLE**) meta-embeddings. To explain this method further let us consider computing the LLE-based meta-embedding, $\boldsymbol{m}_{\text{LLE}}(w)$, of a word $w \in \mathcal{V}_1 \cap \mathcal{V}_2$ using two sources s_1 and s_2. First, they compute the set of nearest neighbours, $\mathcal{N}_j(w)$, of w in s_j and represent w as the linearly-weighted combination of its neighbours by a matrix $\mathbf{A}$ by minimising (13).

$$\underset{\mathbf{A}}{\text{minimise}} \sum_{j=1}^{2} \sum_{w \in \mathcal{V}_1 \cap \mathcal{V}_2} \left|\left| \boldsymbol{s}_j(w) - \sum_{w' \in \mathcal{N}_j(w)} A_{ww'} \boldsymbol{s}_j(w) \right|\right|_2^2 \quad (13)$$

They use AdaGrad to find the optimal $\mathbf{A}$. Next, meta-embeddings are learnt by minimising (14) using the learnt neighbourhood reconstruction weights in $\mathbf{A}$ are preserved in a vector space common to all source embeddings.

$$\sum_{w \in \mathcal{V}_1 \cap \mathcal{V}_2} \left|\left| \boldsymbol{m}_{\text{LLE}}(w) - \sum_{j=1}^{2} \sum_{w' \in \mathcal{N}_j(w)} C_{ww'} \boldsymbol{m}_{\text{LLE}}(w') \right|\right|_2^2 \quad (14)$$

Here, $C_{ww'} = A_{ww'} \sum_{j=1}^{2} \mathbb{I}[w' \in \mathcal{N}_j(w)]$, where $\mathbb{I}$ is the indicator function which returns 1 if the statement evaluated is True. Optimal meta-embeddings can then be found by solving an eigen-decomposition of the matrix $(\mathbf{I}-\mathbf{C})^\top(\mathbf{I}-\mathbf{C})$, where $\mathbf{C}$ is the matrix formed by arranging $C_{ww'}$ as the (w, w') element. This approach has the advantage of not requiring all words to be represented by all sources, thereby obviating the need to predict missing source embeddings prior to meta-embedding.

A.4 Autoencoding (AEME)

Bao and Bollegala (2018) modelled meta-embedding learning as an *autoencoding* problem where information embedded in different sources is integrated at different levels to propose Averaged Autoencoded meta-embedding (**AEME**).

Consider two sources s_1 and s_2, which are encoded respectively by two encoders E_1 and E_2. AEME of w is computed as the ℓ_2 normalised average of the encoded source embeddings as in (15).

$$\boldsymbol{m}_{\text{AEME}}(w) = \frac{E_1(\boldsymbol{s}_1(w)) + E_2(\boldsymbol{s}_2(w))}{||E_1(\boldsymbol{s}_1(w)) + E_2(\boldsymbol{s}_2(w))||_2} \quad (15)$$

Two independent decoders, D_1 and D_2, are trained to reconstruct the two sources from the meta-embedding. E_1, E_2, D_1 and D_2 are jointly learnt to minimise the weighted reconstruction loss given by (16).

$$\underset{E_1,E_2,D_1,D_2}{\text{minimise}} \sum_{w \in \mathcal{V}_1 \cap \mathcal{V}_2} (\lambda_1 \, ||\boldsymbol{s}_1(w) - D_1(E_1(\boldsymbol{s}_1(w)))||_2^2 + \lambda_2 \, ||\boldsymbol{s}_2(w) - D_2(E_2(\boldsymbol{s}_2(w)))||_2^2) \quad (16)$$

The weighting coefficients λ_1 and λ_2 can be used to assign different emphases to reconstruct the two sources and are tuned using a validation dataset. In comparison to methods that learn globally or locally linear transformations (Bollegala et al., 2018; Yin and Schütze, 2016), autoencoders learn nonlinear transformations.

AEME can only use two source embeddings to learn a meta-embedding. Therefore, in cases with more than two source embeddings, we adapt AEME to learn meta-embeddings from meta-embeddings created from two source embeddings and other source embeddings.

B Debiasing Methods

B.1 Hard-debiasing

Bolukbasi et al. (2016) proposed a post-processing approach that projects gender-neutral words to a subspace, which is orthogonal to the gender direction defined by a list of gender-definitional words to reduce the gender stereotypes embedded inside pre-trained word representations. Their hard-debiasing method computes the gender direction as the vector difference between the embeddings of the corresponding gender-definitional words. They denote the n-dimensional pre-trained word embedding of a word w by $\boldsymbol{w} \in \mathbb{R}^n$. W is a set of pre-trained word embeddings $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots \boldsymbol{w}_v$. Here, v is a vocabulary size. To reduce the gender bias, it is assumed that the n-dimensional basis vectors in the $\mathbb{R}^n$ vector space spanned by the pre-trained word embeddings to be $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$. Moreover, without loss of generality, the subspace spanned by the subset of the first $k(< n)$ basis vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_k$ denoted to be $\mathcal{B} \subseteq \mathbb{R}^n$. The projection $\boldsymbol{v}_\mathcal{B}$ of a vector $\boldsymbol{v} \in \mathbb{R}^n$ onto $\mathcal{B}$ can be expressed using the basis vectors as in (17).

$$\boldsymbol{v}_\mathcal{B} = \sum_{j=1}^{k} (\boldsymbol{v}^\top \boldsymbol{b}_j) \boldsymbol{b}_j \quad (17)$$

To show that $\boldsymbol{v} - \boldsymbol{v}_\mathcal{B}$ is orthogonal to $\boldsymbol{v}_\mathcal{B}$ for any $\boldsymbol{v} \in \mathcal{B}$, $\boldsymbol{v} - \boldsymbol{v}_\mathcal{B}$ is expressed using the basis vectors as given in (18).

$$\begin{aligned} \boldsymbol{v} - \boldsymbol{v}_\mathcal{B} &= \sum_{i=1}^{n} (\boldsymbol{v}^\top \boldsymbol{b}_i) \boldsymbol{b}_i - \sum_{j=1}^{k} (\boldsymbol{v}^\top \boldsymbol{b}_j) \boldsymbol{b}_j \\ &= \sum_{i=k+1}^{n} (\boldsymbol{v}^\top \boldsymbol{b}_i) \boldsymbol{b}_i \end{aligned} \quad (18)$$

It can be seen that there are no basis vectors in common between the summations in (17) and (18). Therefore, ${\boldsymbol{v}_{\mathcal{B}}}^\top(\boldsymbol{v} - \boldsymbol{v}_{\mathcal{B}}) = 0$ for $\forall \boldsymbol{v} \in \mathcal{B}$.

To identify gender subspace, a set of n a masculine and feminine word pairs $D_1, D_2, ..., D_n \subset W^2$ is defined, here each pair of words indicates gender. The average vector μ of the defining sets is represented in (19).

$$\mu_i := \sum_{\boldsymbol{w} \in D_i} \frac{\boldsymbol{w}}{|D_i|} \tag{19}$$

Let the bias subspace $\mathcal{B}$ be the first k rows of Singular Value Decomposition SVD ($\mathbf{C}$),

$$\mathbf{C} := \sum_{i=1}^{n} \sum_{\boldsymbol{w} \in D_i} (\boldsymbol{w} - \mu_i)^\top (\boldsymbol{w} - \mu_i)/|D_i| \tag{20}$$

Hard-debiasing removes the bias by zero projection of all neutral words into the bias subspace $\mathcal{B}$. Then, debiased embedding $\boldsymbol{d}_{\text{hard}}(\boldsymbol{w})$ is represented in (21):

$$\boldsymbol{d}_{\text{hard}}(\boldsymbol{w}) := \frac{\boldsymbol{w} - \boldsymbol{w}_{\mathcal{B}}}{||\boldsymbol{w} - \boldsymbol{w}_{\mathcal{B}}||} \tag{21}$$

Gonen and Goldberg (2019) showed that hard-debiasing does not completely remove gender biases from embeddings. On the other hand, the motivation of Single-Source Multi-Debiasing is to complement weaknesses of each debiasing method by combining the various debiasing methods. Therefore, we use hard-debiasing, a method known to produce incomplete debiasing, to investigate whether we could overcome its limitations by meta-embedding with source embeddings produced by other debiasing methods.

B.2 Iterative Null-space Projection (INLP)

INLP was proposed by Ravfogel et al. (2020) to remove bias in pre-trained embeddings by iteratively projecting onto null-space. They train multiple linear classifiers to detect bias in a pre-trained embedding and remove information by projecting the embedding into the null space of the weights of each linear classifier. By adapting multiple classifiers, it is possible to remove bias by projecting embeddings into the null space using dozens of directions based on the data.

First, let C be the parameter of the linear classifier that detects the bias in the word embeddings. For example, in gender bias detection, this linear classifier will classify whether the representation is feminine or masculine. The fact that the classifier cannot classify the embedding as feminine or masculine, as in (22), means that there is no bias in the embedding.

$$C(P_{N(C)}\boldsymbol{w}) = 0 \quad \forall \boldsymbol{w} \tag{22}$$

Here, W is projected to a space orthogonal to C, i.e., null-space, so that the decision boundary of C cannot detect the bias. The null-space at C is defined as $N(C) = \{\boldsymbol{w}|C\boldsymbol{w} = 0\}$, and the projection matrix onto $N(C)$ is $P_{N(C)}$.

Relations in a multidimensional space can be captured in multiple linear directions (hyperplanes). Therefore, it is not sufficient to project the embedding into the null space of a single linear classifier. To solve this problem, they adapt the classifier iteratively. The projection matrix P is iterated m times as $P = P_{N(C_m)}P_{N(C_{m-1})} \dots P_{N(C_1)}$. C_i is learned from W_{i-1} projected into the null space of C_{i-1}. Debiased embedding $\boldsymbol{d}_{\text{inlp}}(\boldsymbol{w})$ is represented in (23):

$$\boldsymbol{d}_{\text{inlp}}(\boldsymbol{w}) := P\boldsymbol{w} \tag{23}$$

B.3 Dict-debiasing

Kaneko and Bollegala (2021b) proposed dict-debiasing – a method for debiasing pre-trained word embeddings using dictionary definitions. This method does not need the types of biases to be pre-defined in the form of word lists and learns the constraints that must be satisfied by unbiased word embeddings automatically from dictionary definitions of the words.

This method assumes that a dictionary $\mathcal{D}$ containing the definition, $g(w)$ of w, is given. If the pre-trained embeddings distinguish among the different senses of w, then the gloss for the corresponding sense of w in the dictionary can be used as $g(w)$. Given $\boldsymbol{w}$, which is the word embedding of a word w, they model the debiasing process as the task of learning an encoder $E(\boldsymbol{w}; \boldsymbol{\theta}_e)$ that returns an $m(\leq n)$-dimensional debiased version of $\boldsymbol{w}$. To preserve the dimensionality of the input embeddings, they set $m = n$.

To preserve semantic information during the debiasing process, they decode the encoded version of $\boldsymbol{w}$ using a decoder D_c parametrised by $\boldsymbol{\theta}_c$ and define J_c to be the reconstruction loss given by (24).

$$J_c(w) = ||\boldsymbol{w} - D_c(E(\boldsymbol{w}; \boldsymbol{\theta}_e); \boldsymbol{\theta}_c)||_2^2 \tag{24}$$

To ensure that the encoded version of $\boldsymbol{w}$ is similar to $g(w)$, $g(w)$ is represented by a sentence embedding vector $\boldsymbol{g}(w) \in \mathbb{R}^n$. For the simplicity, they use the smoothed inverse frequency (SIF; Arora et al., 2017) for creating $\boldsymbol{g}(w)$. SIF computes the embedding of a sentence as the weighted average of the pre-trained word embeddings of the words in the sentence, where the weights are computed as the inverse unigram probability. Next, the first principal component vector of the sentence embeddings is removed. The dimensionality of the sentence embeddings created using SIF is equal to that of the pre-trained word embeddings used. Therefore, both $\boldsymbol{w}$ and $\boldsymbol{g}(w)$ are in the same n-dimensional vector space. The debiased embedding $E(\boldsymbol{w};\boldsymbol{\theta}_e)$ of w are decoded using a decoder D_d, parametrised by $\boldsymbol{\theta}_d$. The squared ℓ_2 distance between decoded embedding and $\boldsymbol{g}(w)$ is computed to define an objective J_d given by (25).

$$J_d(w) = ||\boldsymbol{g}(w) - D_d(E(\boldsymbol{w};\boldsymbol{\theta}_e);\boldsymbol{\theta}_d)||_2^2 \tag{25}$$

To remove unfair biases from pre-trained word embedding $\boldsymbol{w}$ of a word w, it is projected into a subspace that is orthogonal to the dictionary definition vector $\boldsymbol{g}(w)$. This projection is denoted by $\phi(\boldsymbol{w}, \boldsymbol{g}(w)) \in \mathbb{R}^n$. The debiased word embedding, $E(\boldsymbol{w};\boldsymbol{\theta}_e)$, must be orthogonal to $\phi(\boldsymbol{w}, \boldsymbol{g}(w))$, and this is formalised as the minimisation of the squared inner-product given in (26).

$$J_a(w) = \left(E(\phi(\boldsymbol{w}, \boldsymbol{g}(w)); \boldsymbol{\theta}_e)^\top E(\boldsymbol{w};\boldsymbol{\theta}_e)\right)^2 \tag{26}$$

Note that because $\phi(\boldsymbol{w}, \boldsymbol{g}(w))$ lives in the space spanned by the original (prior to encoding) vector space, it must be first encoded using E before considering the orthogonality requirement.

To derive $\phi(\boldsymbol{w}, \boldsymbol{g}(w))$, (17) and (18) are used. Considering that $\boldsymbol{s}(w)$ defines a direction that does not contain any unfair biases, the vector rejection of $\boldsymbol{w}$ on $\boldsymbol{g}(w)$ can be computed following this result.[12] Specifically, by subtracting the projection of $\boldsymbol{w}$ along the unit vector defining the direction of $\boldsymbol{g}(w)$ to compute ϕ as in (27).

$$\phi(\boldsymbol{w}, \boldsymbol{s}(w)) = \boldsymbol{w} - \boldsymbol{w}^\top \boldsymbol{g}(w) \frac{\boldsymbol{g}(w)}{||\boldsymbol{g}(w)||} \tag{27}$$

The linearly-weighted sum of the above-defined three objective functions is considered as the total objective function as given in (28).

$$J(w) = \alpha J_c(w) + \beta J_d(w) + \gamma J_a(w) \tag{28}$$

Here, $\alpha, \beta, \gamma \geq 0$ are scalar coefficients satisfying $\alpha + \beta + \gamma = 1$. Finally, debiased embedding $\boldsymbol{d}_{\text{dict}}(\boldsymbol{w})$ is represented in (29):

$$\boldsymbol{d}_{\text{dict}}(\boldsymbol{w}) := E(\boldsymbol{w};\boldsymbol{\theta}_e) \tag{29}$$

[12] The rejection of a vector $\boldsymbol{b}$ by another vector $\boldsymbol{a}$ is defined as $\boldsymbol{a} - (\boldsymbol{a}^\top \boldsymbol{b})\boldsymbol{a}$.