Bias Invariant Approaches for Improving Word Embedding Fairness

Siyu Liao
liasiyu@amazon.com
Amazon.com
Seattle, WA, USA

Rongting Zhang
rongtz@amazon.com
Amazon.com
Seattle, WA, USA

Barbara Poblete
bpoblete@uchile.cl
University of Chile, Chile
Amazon.com, USA

Vanessa Murdock
vmurdock@amazon.com
Amazon.com
Seattle, WA, USA

ABSTRACT

Many public pre-trained word embeddings have been shown to encode different types of biases. Embeddings are often obtained from training on large pre-existing corpora, and therefore resulting biases can be a reflection of unfair representations in the original data. Bias, in this scenario, is a challenging problem since current mitigation techniques require knowing and understanding existing biases in the embedding, which is not always possible. In this work, we propose to improve word embedding fairness by borrowing methods from the field of data privacy. The idea behind this approach is to treat bias as if it were a special type of training data leakage. This has the unique advantage of not requiring prior knowledge of potential biases in word embeddings. We investigated two types of privacy algorithms, and measured their effect on bias using four different metrics. To investigate techniques from differential privacy, we applied Gaussian perturbation to public pre-trained word embeddings. To investigate noiseless privacy, we applied vector quantization during training. Experiments show that both approaches improve fairness for commonly used embeddings, and additionally, noiseless privacy techniques reduce the size of the resulting embedding representation.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Security and privacy** → **Usability in security and privacy**.

KEYWORDS

Word Embedding; Fairness; Privacy

ACM Reference Format:
Siyu Liao, Rongting Zhang, Barbara Poblete, and Vanessa Murdock. 2023. Bias Invariant Approaches for Improving Word Embedding Fairness. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3614792

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614792

1 INTRODUCTION

For many natural language technology tasks, word embeddings are a fundamental tool for machines to understand textual data. A word embedding is a real-valued vector representing a single word. The vectors are semantically meaningful, as words that have a similar meaning are close to each other in the vector space [16]. Relationships between pairs of words can be directly computed from their embedding vectors with vector operations. As an example, vector difference has been shown effective in the analogy completion task [32], where the most well-known example is *king – man + woman = queen*: the embedding vector for "man" subtracted from the embedding vector for "king", added to the embedding vector for "woman" yields the embedding vector for "queen".

Remarkably, word embedding vectors need not be trained specifically for a given task, rather they are trained on large general corpora. As might be expected, the resulting word embeddings also capture stereotypical relations between words due to bias in the corpus data, as well as other factors, including how written language is used. Bias in terms of inferred associations based on gender, race, religion and other demographic attributes are well documented [29]. In particular, this paper conceptualizes "bias" and "fairness" in the word embedding space, which is based on word associations. The conceptualization is self-evident and observable in experiments [7]. Other topics such as hypothesis-only bias or inductive bias are beyond the scope of this paper. Similar to Sesari et al. [42], we define bias as the absence of fairness in word embedding.

Word embeddings are applied to many language technologies, such as machine translation [28], text classification [44] and information retrieval [35]. Given the widespread real-world applications derived from such tasks, there is a pressing need to improve fairness in the embedding representations. This is evidenced by the growing body of work devoted to identifying, measuring and reducing bias in word embeddings [2, 8, 9, 29]. An important limitation to current approaches to mitigating bias is that they depend on identifying the type or target of the bias upfront, or they require background knowledge of possible problematic or unfair representations encoded in the embeddings. As a result, bias mitigation can only be addressed for well-known misrepresentation stereotypes or as a reactive measure once a new type of bias is discovered.

For instance, an approach for debiasing embeddings presented by Bolukbasi et al. [8], is based on the idea that since words are represented in a vector space, bias therefore exists within a certain

subspace. So, by using several word vectors that are representative of a particular bias, it should be possible to reconstruct the corresponding bias subspace. For example, the gender space can be found as the top component of word embedding vectors of gendered terms such as pronouns and nouns. Therefore, subtracting the projection into the gender space should yield a more gender neutral embedding representation of a text. This requires a defined target (gender bias) known in advance, and addressing gender bias in this way will not serve to reduce other types of bias. Since it is not possible to identify all targets of bias, what is needed is a method that reduces bias without prior knowledge of its specific nature.

In this work we propose a bias invariant algorithm to improve word embedding fairness, borrowing from the field of differential privacy, which is unsupervised and does not rely on predefined lists of biased words. In particular, we investigate two kinds of privacy preserving algorithms: noise-based differential privacy and noiseless privacy algorithms. We propose a heuristic to apply Gaussian noise perturbation to word embeddings, and we give a theoretical proof of how this algorithm improves fairness. We demonstrate the solution can be applied regardless of the types of bias in the training corpus, thus these algorithms serve as a general approach to improving fairness. We demonstrate the improvements to fairness yielded by privacy protection techniques, and explore the trade-off between fairness and effectiveness of word embeddings debiased in this way.

Our experiments show that the application of differential privacy algorithm improves word embedding fairness across the board for all metrics. For the metrics WEAT and WEAT-ES, the difference can be substantial (improvements of 200% - 500%). This improvement comes at the cost of a 2%+ decrease in accuracy and F-score in the downstream tasks of chunking, NER, and POS tagging, in particular for ConceptNet [43] and Fasttext [19] embeddings. On the other hand, noiseless privacy algorithms improve fairness across the board, with statistically significant improvements in performance for both Wikipedia and MS Marco data, for all downstream tasks. Overall, by applying principles from privacy preservation we show that it is possible to improve fairness in word embeddings in an unsupervised and bias-invariant manner. In this regard, we find that noise-based privacy algorithms appear to be more efficient, since they are applied as a post-processing step. On the other hand, noiseless privacy algorithms offer the best trade-off between improving fairness and maintaining effectiveness in downstream tasks.

This paper is organized as follows: Section 2 introduces related work in word embeddings, fairness metrics and privacy algorithms. Section 3 presents our approaches based on two algorithms borrowed from differential privacy, and noiseless privacy. Section 4 details the experimental set up, and Section 5 presents results on the resulting embedding fairness, and their usefulness in downstream tasks. Section 6 discusses our findings and Section 7 presents our conclusions.

2 RELATED WORK

We present a brief overview of two relevant research areas related to our current work: 1) bias in word embeddings and 2) privacy preserving methods. Although, our work is not in the area of privacy preservation, we do borrow and adapt two approaches for the purpose of improving word embedding fairness.

Bias in word embeddings

Describing documents with vectors of words has long been used by libraries to make their collection searchable by subject. As libraries became digitized, the Vector Space Model [37] was used to represent documents as vectors of discrete index terms. In this model, the similarity between documents (or between a document and a search query) is measured by the cosine similarity between document vectors. More recently, this idea was extended to continuous valued vectors [31] [5], called word embeddings, which were used as input to neural models, or used directly to represent semantic relationships between words in language tasks. Word embeddings have become fundamental building blocks for many language technologies, and support many common everyday tasks. For this reason, evidence showing [29] that embeddings trained on large online corpora display significant biases has generated important societal concerns. Some examples of biases found to be encoded in embeddings are gender, ethnic, and religious.

The Word Embedding Association Test (WEAT) [9] is a standard fairness measurement metric inspired by the Implicit Association Test (IAT) [20]. In the IAT, people are asked to link an attribute ("happy") to a target ("puppy") to indicate an association, and their response time is measured. *Target words* describe a specific group of people, and *attribute words* describe attributes of a social group. For word embeddings, WEAT measures the cosine similarity between a predefined set of attribute words and targets words.

Bolukbasi et al. [8] showed that word embedding bias encode gender stereotypes in vector subspaces. In particular, they show that the subspace can be determined by a predefined list of words, and subtracting the projection in this space should neutralize bias in the word embedding vectors. Such projection can be directly built from the list or learned towards minimizing the effect on overall word embedding, which are called hard and soft debiasing, respectively. However, this method is limited to binary stereotypes like those related to man and woman, not capturing more nuanced language that contains multiple different biases. A straightforward solution to extend the idea to multi-class stereotypes is to subtract multiple components from principle component analysis (PCA) using predefined lists of words for all classes [29]. However, this begs the question of how to comprehensively and accurately curate lists of terms, as well as to how to define classes of stereotypes.

Yang and Liu [48] assume the relation between stop words and content words can be described in a half-sibling regression (HSR) structure [41] in causal inference. Although they aim at removing noise in word embeddings, their approach is similar to soft debiasing but with a pre-defined list of stop words. Another approach [24] generalizes the idea of Bolukbasi et al. [8] into a multi-objective optimization process. It considers the issue of hard debiasing failing to maintain the relation between gender neutral words and their neighbors. The problem is formulated to minimize gender biased illicit association [18], semantic meaning loss, and gender bias, which are referred as repulsion, attraction and neutralization (RAN), respectively.

Besides these post-processing algorithms, there are also methods proposed to improve fairness during word embedding training. For example, Hube et al. [23] reduce stereotypes in people names at training time. In their work, an oracle classifier is required to help identify the bias, and more importantly, the fairness improvement is limited to the specific list of names. In a similar approach, Zhao et al. [49] train word embeddings with a data augmentation method. The method builds a dataset specifically to address gender stereotypes, where all gender pronouns are linked to stereotypical entities such as occupations.

The above mentioned methods are limited to specific stereotypes due to their need for pre-defined lists of words. This has the disadvantage that the algorithm designer must come up with the lists of words for each class, which is itself a source of bias. Moreover, the resulting word embeddings can still be vulnerable to biases that are unknown to the algorithm designer. The challenge of improving word embedding fairness requires addressing bias in the data (including in the pre-defined lists of words) and the algorithm design itself. By contrast, the methods presented in this paper are based on ideas from differential and noiseless privacy. Since they are not dependent on pre-defined words, the debiasing process is not impacted by this additional source of bias. Further, as we present general algorithms, the target of the bias (gender, ethnicity, etc.) need not be known.

Privacy preserving methods

Differential privacy has been found to be an useful tool to prevent data leakage in machine learning models [1], making it difficult for an adversary to infer the original training data, given a trained model. Classical differential privacy is defined on discrete databases, where an adversary finds data leakage by measuring the difference between two similar queries [14]. For example, when allowed to get XOR results of records in a Boolean database, an adversary can compare the difference between the top 99 rows and the top 100 rows in order to get the bit value for 100-th row. The essential goal of differential privacy algorithms is ensuring that a data record cannot be distinguished by an adversary.

Differential privacy algorithms do not promise an adversary cannot find any correlation between the data record and the statistics summarized from the database [14]. For example, an individual record may or may not exist in the database, but an adversary can still draw some conclusions from an analysis of the database, and apply the conclusions to an individual record. Current differentially private algorithms typically add randomness (noise) to the results, to make it hard for an adversary to gain information from analyzing differences in results. These algorithms assume the adversary knows almost all records but one, so the adversary can identify the record information by analyzing the output difference given a set of input queries. If the adversary only knows a fraction of the records, the rest will look like random variables. This has been successfully exploited to get "free" privacy protection, to generalize to the concept of noiseless privacy [6, 13, 21]. A formal theoretical framework built on top of noiseless privacy preserving algorithms can be found in [15]. The basic idea behind noiseless privacy is to restrict the output values so that they are limited to a smaller, indistinguishable set of values.

3 BIAS INVARIANT APPROACH TO IMPROVE FAIRNESS

In this section, we present two bias invariant approaches for improving word embedding fairness. These methods are inspired by privacy preservation algorithms. We first present a post-processing algorithm for debiasing pre-trained word embeddings using Gaussian perturbation, along with a theoretical proof that this improves word embedding fairness via differential privacy. Second, we present a method based on a noiseless privacy approach using vector quantization to improve fairness for word embeddings trained from scratch.

3.1 Borrowing from Differential Privacy and Noiseless Privacy

Generally speaking, differential privacy is a mechanism that guarantees for each individual record in a statistical database, that the output of a differentially private analysis will be indistinguishable, regardless of whether the record exists or does not exist in the data. A discussion of differential privacy in general is beyond the scope of this paper, but an overview of the concepts can be found in Dwork et al. [14] and Grining and Klonowski [21].

In our work, we propose that differential privacy techniques can guarantee that word embeddings derived from two corpora that differ in a single word, will be indistinguishable from each other. Therefore, a target word (e.g., "homemaker") should not be closer to a specific attribute word (e.g., "woman") than to another (e.g., "man"), since distances remain similar whether the target word is or is not in the corpus. This follows a similar principle to the definition of *differentially private deep learning* by Abadi et al. [1].

More formally, given two neighboring corpora C and C' that differ in one word (the word frequency between C and C' differs by one), an algorithm $\mathcal{A} : C \rightarrow \mathbb{R}^d$ taking an input corpus and generating an output word embedding is (ϵ, δ)-differentially private, if for every pair of neighboring corpora and any possible output embedding $\in \mathbb{R}^d$, the probability of the occurrence of the embedding is within a privacy budget (ϵ, δ). Specifically:

$$\forall i \in \{1, \ldots, n\}, \;\; \mathbb{P}[\mathbf{w}_i = \mathbf{w}] \leq e^{\epsilon} \mathbb{P}[\mathbf{w}'_i = \mathbf{w}] + \delta, \tag{1}$$

where ϵ and δ are the privacy budget. $\mathbf{w}_i$ and $\mathbf{w}'_i$ are output embedding from corpus C and C', respectively.

One limitation of differential privacy algorithms is that they achieve randomness by adding noise to the data, which decreases the data utility in downstream tasks. Rather than add noise to the data, noiseless privacy restricts the output to a smaller set of values. Intuitively, an adversary can only see a limited number of output values, and it becomes hard to gain information by analyzing the output difference.

More formally, an algorithm $\mathcal{A} : C \rightarrow \mathcal{R}^d$ taking an input corpus C and generating an output word embedding w is ϵ-noiselessly private, if for any neighboring corpus C', the cardinality of the output embedding w' is bounded:

$$\forall i \in \{1, \ldots, n\}, \;\; |\{\mathbf{w}'_i\}| < 2^{\epsilon}, \tag{2}$$

where $|\cdot|$ returns the set cardinality and ϵ is the privacy budget. In other words, for any neighboring input, the output attains only a small set of values. For example, in a binary representation, each

dimension is only limited to a value of 0 or 1, and $|\{\mathbf{w}_i'\}| < 2^d$. To exclude the factor of the vector dimension, in our experiments we simply define the privacy budget using the bit representation (for the 2-bit representation, $\epsilon = 2$, and for the 1-bit representation $\epsilon = 1$).

3.2 Fairness Improvement using Gaussian Perturbation

Adding random Gaussian noise $\mathbf{n} \sim N(0, \sigma^2\mathbf{I})$ is a standard way to achieve differential privacy [14]. The value of σ is constrained by the privacy budget (ϵ, δ) and also algorithmic sensitivity $\Delta\mathcal{A}$:

$$\Delta\mathcal{A} = \max_{C,C'} ||\mathcal{A}(C) - \mathcal{A}(C')||_2. \tag{3}$$

More specifically, σ can be determined by a function of $\epsilon, \delta, \Delta\mathcal{A}$. Here we did not use the classical analytical result [14], i.e., $\sigma = \Delta\mathcal{A}\sqrt{2\log(1.25/\delta)}/\epsilon$ as its privacy proof requires $\epsilon < 1$ and this restricts the privacy budget in a way that might not be ideal for a given application. In experiments, to compute of the degree of Gaussian perturbation with a given privacy budget, we adopted a modified algorithm [4] to calculate σ given ϵ, δ and $\Delta\mathcal{A}$.

3.2.1 Word Embedding Sensitivity Heuristic. As it is not straightforward to compute an accurate sensitivity for training word embeddings, we provide an heuristic to estimate the sensitivity based on the embedding point-wise mutual information (PMI). That is,

$$\Delta\mathcal{A} = \log 2/2M \tag{4}$$

where M is the Euclidean norm of the maximum length embedding $\max_i ||\mathbf{w}_i||_2$. The related derivation is in the Appendix.

3.2.2 Fairness improvement. Given a privacy budget and algorithm sensitivity, we can generate a differentially private word embedding. In this section we present a proof that this algorithm can improve the cosine similarity-based fairness metrics.

Proposition 3.1. *Assume we have two embedding vectors $v_1, v_2 \in \mathbb{R}^n$ and noise vectors ϵ_1, ϵ_2 sampled independently from the same n-dimensional spherically symmetric distribution D. The expected cosine similairty between the two embedding vectors with noise, $v_1 + \epsilon_1$ and $v_2 + \epsilon_2$, equals the product of the cosine similarity between the original vectors and a shrinking factor dependent on their norms. Specifically,*

$$\mathbb{E}[\cos(v1 + \epsilon_1, v2 + \epsilon_2)] = \alpha(\|v_1\|, \|v_2\|)\cos(v1, v2) \tag{5}$$

where $0 < \alpha < 1$.

Since an n-dimensional Gaussian distribution is spherically symmetric, we obtain the following conclusion as a special case of Proposition 3.1.

Corollary 3.2. *Gaussian perturbation-based differentially private word embeddings will be less biased under cosine similarity metrics.*

Proof. As the random noise vectors are sampled independently, it is enough to show for noise vector ϵ sampled from distribution D that the expected cosine similarity between v_1 and $v_2 + \epsilon$ equals the product of the cosine similarity between the original vectors and a shrinking factor dependent on the norm of v_2. More specifically,

$$\mathbb{E}[\cos(v_1, v_2 + \epsilon)] = \alpha(\|v_2\|)\cos(v_1, v_2), \tag{6}$$

where $0 < \alpha < 1$. Without loss of generality, we assume that $\|v_1\| = 1$. Since ϵ is sampled from a spherical symmetric distribution, the expectation can be written as

$$\begin{aligned}\mathbb{E}[\cos(v_1, v_2 + \epsilon)] &= \int_0^\infty \frac{1}{V_n}\int_{S_n} \cos(v_1, v_2 + rx)dxh(r)dr \\ &= \int_0^\infty \frac{1}{V_n}\int_{S_n} \cos(v_1, \frac{v_2}{\|v_2\|} + \frac{r}{\|v_2\|}x)dxh(r)dr\end{aligned}$$

where V_n is the surface area of the n-dimensional unit sphere S_n and $\int_0^\infty h(r)dr = 1, h > 0$. For the inner integral, following Lemma .1 and the fact that $\cos(v_1, \frac{v_2}{\|v_2\|}) = \cos(v_1, v_2)$, we obtain

$$\frac{1}{V_n}\int_{S_n} \cos(v_1, \frac{v_2}{\|v_2\|} + \frac{r}{\|v_2\|}x)dx = f(\frac{r}{\|v_2\|})\cos(v_1, v_2),$$

with $0 < f(\frac{r}{\|v_2\|}) < 1$. Thus, we have

$$\begin{aligned}\mathbb{E}[\cos(v_1, v_2 + \epsilon)] &= \int_0^\infty f(\frac{r}{\|v_2\|})\cos(v_1, v_2)h(r)dr \\ &= \alpha(\|v_2\|)\cos(v_1, v_2),\end{aligned}$$

and $0 < \alpha(\|v_2\|) = \int_0^\infty f(\frac{r}{\|v_2\|})h(r)dr < 1$.

□

We note that sometimes the fairness metric is based on differences of cosine similarities between sets of vectors (e.g. WEAT). In such case, a decrease in metric is expected as long as norms of vectors in consideration do not differ significantly. Thus, we conclude differentially private word embedding can improve cosine similarity-based fairness metric.

3.3 Fairness Improvement using Quantization

In noiseless privacy, uniform quantization has been shown to have privacy guarantees [15]. Quantization is also used in deep learning to represent model parameters using fewer bits to compress large models [27]. We describe the quantization algorithm in this section. Formally, let Q divide the input value domain into equal sized intervals. We define $q(\cdot)$ as the step function for quantization, where

$$q(x) = \sum_i \theta_i g_i(x)$$

θ_i is the quantization value and $g_i(\cdot)$ is disjoint indicator functions for the interval i. For example, with $Q = 1$ dividing input domain $[-1, 1]$ into two intervals $[-1, 0]$ and $(0, 1]$, we have

$$q(x) = (-1) \cdot \mathbb{1}\{x \in [-1, 0]\} + 1 \cdot \mathbb{1}\{x \in (0, 1]\},$$

which is equivalent to the sign function on $[-1, 1]$. During the training process, the typical practice is to maintain model parameters in both full precision (e.g., 32 bits single precision floating point) and quantized precision. Let $\mathbf{W}$ be the word embedding parameter, and then $\mathbf{W}$ is updated at each training iteration as following:

$$\mathbf{W} \leftarrow \mathbf{W} - \alpha \cdot \nabla L(q(\mathbf{W})), \tag{7}$$

where ∇L is the gradient with quantized parameters $q(\mathbf{W})$ as input, and α is the learning rate. It's noted that in this update step, both full precision and quantized precision are involved in the computation. When the training is complete, the quantized word embedding can be used for deployment. Theoretically, it has been found that

Table 1: Word embedding privacy budget configurations for Gaussian perturbation and quantization.

Word Embedding	Dim.	Configuration
conceptnet-1	300	$\epsilon = 1.00, \delta = 1e-03, N(0, 0.45)$
conceptnet-2	300	$\epsilon = 1.00, \delta = 1e-05, N(0, 0.65)$
conceptnet-3	300	$\epsilon = 0.10, \delta = 1e-03, N(0, 3.02)$
conceptnet-4	300	$\epsilon = 0.10, \delta = 1e-05, N(0, 5.33)$
fasttext-wiki-1	300	$\epsilon = 1.00, \delta = 1e-03, N(0, 0.11)$
fasttext-wiki-2	300	$\epsilon = 1.00, \delta = 1e-05, N(0, 0.16)$
fasttext-wiki-3	300	$\epsilon = 0.10, \delta = 1e-03, N(0, 0.75)$
fasttext-wiki-4	300	$\epsilon = 0.10, \delta = 1e-05, N(0, 1.33)$
glove-twitter-1	200	$\epsilon = 1.00, \delta = 1e-03, N(0, 0.05)$
glove-twitter-2	200	$\epsilon = 1.00, \delta = 1e-05, N(0, 0.08)$
glove-twitter-3	200	$\epsilon = 0.10, \delta = 1e-03, N(0, 0.35)$
glove-twitter-4	200	$\epsilon = 0.10, \delta = 1e-05, N(0, 0.63)$
glove-wiki-1	300	$\epsilon = 1.00, \delta = 1e-03, N(0, 0.05)$
glove-wiki-2	300	$\epsilon = 1.00, \delta = 1e-05, N(0, 0.07)$
glove-wiki-3	300	$\epsilon = 0.10, \delta = 1e-03, N(0, 0.34)$
glove-wiki-4	300	$\epsilon = 0.10, \delta = 1e-05, N(0, 0.59)$
lexvec-1	300	$\epsilon = 1.00, \delta = 1e-03, N(0, 0.03)$
lexvec-2	300	$\epsilon = 1.00, \delta = 1e-05, N(0, 0.04)$
lexvec-3	300	$\epsilon = 0.10, \delta = 1e-03, N(0, 0.21)$
lexvec-4	300	$\epsilon = 0.10, \delta = 1e-05, N(0, 0.37)$
word2vec-1	300	$\epsilon = 1.00, \delta = 1e-03, N(0, 0.04)$
word2vec-2	300	$\epsilon = 1.00, \delta = 1e-05, N(0, 0.06)$
word2vec-3	300	$\epsilon = 0.10, \delta = 1e-03, N(0, 0.27)$
word2vec-4	300	$\epsilon = 0.10, \delta = 1e-05, N(0, 0.48)$
w2b-wiki	800	single-precision floating-point
w2b-wiki-1b	800	1 bit precision
w2b-wiki-2b	800	2 bits precision
w2b-marco	128	single-precision floating-point
w2b-marco-1b	128	1 bit precision
w2b-marco-2b	128	2 bits precision

quantization is equivalent to proximal gradient method with a structured regularization term [3].

4 EXPERIMENTAL SETUP

We evaluate word embedding fairness with and without privacy, using the evaluation framework WEFE [2]. In addition to measuring the fairness of the word embeddings themselves, we evaluate the effect of privacy on the utility of the embeddings in the following downstream tasks: Part-of-Speech (POS) tagging on the Penn Treebank [30], chunking from CoNLL'00 [38] and named-entity recognition (NER) from CoNLL'03 [39]. For each downstream task, the average result is reported over 5 random runs. Since word lists and baseline embeddings are always fixed when evaluating fairness, we performed a Wilcoxon rank sum test (as suggested in Dror et al. [12]) only in downstream tasks when comparing models trained on the experimental embedding and models trained on the original word embedding.

WEFE [2] measures word embeddings on several fairness metrics for different social biases (ethnicity, gender and religion). To measure the overall fairness under each metric, it calculates the weighted mean of social biases using a pre-defined word list of known bias terms. The mean is weighted by the length of the word list. The algorithms presented in Section 3 are not designed to capture specific forms of social bias, rather they are designed to capture *any* bias. We use WEFE as it is the current standard method of measurement, but we report the overall fairness measurement for each metric, rather than the metrics for each type of bias.

WEFE reports four different metrics (WEAT, WEAT-ES, RND and RNSB) for each type of social bias. WEAT [9] is a popular fairness metric calculating the difference of the distance between an attribute word (e.g. "programmer") and two target words (e.g., "man" and "woman"); 2) WEAT-ES is the result of WEAT divided by the standard deviation; 3) RND [17] calculates difference between the distance of the average of two sets of target words against the same attribute word; 4) RNSB [45] measures the difference of a binary logistic regression classifier's output probability with input target words, where the classification model is trained on two sets of attribute words as positive and negative examples, respectively. It should be noted that for all reported fairness metrics, a lower metric value indicates less difference between target words, and therefore a more fair embedding.

For Gaussian perturbation, we choose the classical word embedding vectors as reported by WEFE. We used the following pre-trained embeddings: ConceptNet knowledge graph embeddings [43], FastText embeddings trained on Wikipedia [19], Glove embeddings trained on Twitter and Wikipedia [34], LexVec trained on Common Crawl data [36] and the classical Word2vec trained on Google News [31]. In addition, we also compare with publicly available debiased embeddings, i.e., gender hard debiased Word2vec embedding [8], Repulsion, Attraction, and Neutralization (RAN) debiased Glove embedding [24], and Half Sibling Regression (HSR) denoised word2vec embedding [48].

For noiseless privacy with word embedding quantization, we compare the word embeddings trained on the same corpus with and without quantization. Word embeddings without quantization are represented in full precision (32-bit single-precision floating-point format). Quantized word embeddings are configured with 1-bit and 2-bit representations. We adopt the implementation of the quantization method, and the pre-trained word embedding on 2017 English Wikipedia from Lam [25]. We also trained the full precision and quantized word embeddings from MS Marco Conversational Search data [33], using all queries from this data. Note that this query corpus is distinct from other corpora typically used for word embeddings. Search queries are usually short (typically 2 - 6 words), composed of nouns and adjectives with little or no semantic context, and in this data, they relate to voice search (rather than to current events, or other informational topics).

Table 1 lists all configuration details in the experiments. For Gaussian perturbation, we explore the privacy budget with $\epsilon \in \{1.0, 0.1\}, \delta \in \{1e-3, 1e-5\}$. In the table, for readability when we refer to a particular configuration, we number the base embedding source (i.e. conceptnet-1, conceptnet-2, etc.). The quantized word embeddings are named with a suffix indicating the bit representation. For example, w2b-wiki-1b is the 1-bit representation of the word embeddings trained from Wikipedia data, and w2b-wiki is the full precision representation.

5 RESULTS

This section lists all experimental results for fairness evaluation and downstream task evaluation. Table 2 shows results for differential privacy and noiseless privacy. The baseline (original) word embeddings are presented in italics, and the most fair (lowest value) results are in boldface. We indicate statistically significant results ($p < 0.05$) with respect to the baseline original embedding with a †, according to a Wilcoxon rank sum test.

5.1 Fairness Evaluation

As described above, we evaluate word embedding fairness using WEFE. All word embedding vectors are measured using the same metrics and word lists. Table 2 shows that Gaussian perturbation improves word embedding fairness across the board, with the exception of ConceptNet evaluated with RND and RNSB. A smaller privacy budget (with larger Gaussian perturbation) results in a more fair word embedding in terms of both WEAT and WEAT-ES, regardless of the data source. Notably, compared with other word list-based debiasing methods, Gaussian perturbation is a simple and effective algorithm to improve word embedding fairness. The hard debiasing and RAN methods (word2vec and glove-wiki respectively) outperforms the other methods, according to hte RND and RNSB metrics.

Table 2 shows that quantization improves fairness across all metrics. The improvement on WEAT and WEAT-ES is not as great as on RND and RNSB. In particular, for RND and RNSB, the lower precision representation is always better than full precision.

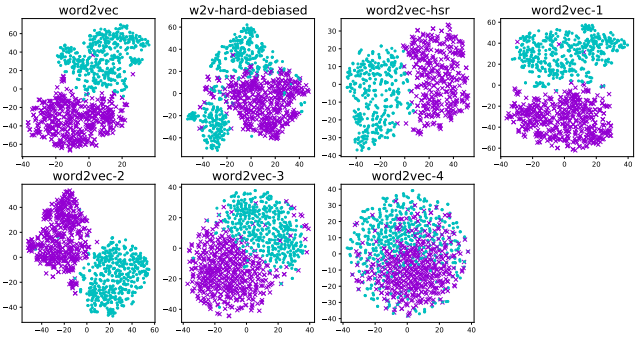
5.2 Downstream Task Evaluation

Table 2 compares the effectiveness of the word embeddings on common downstream tasks. As in Chiu et al. [10], accuracy is reported for Part-of-Speech tagging on the Penn Treebank, and F1-scores are reported for named entity recognition (NER) on data from CoNLL'03 and chunking on data from CoNLL'00.

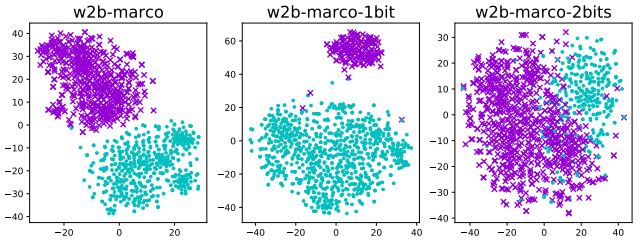
Table 2 shows that adding noise via Gaussian perturbation generally results in a performance drop compared with original word embedding. This is intuitive, as it would be expected that adding noise to the data would decrease the accuracy. There are a few exceptions, for example, Gaussian perturbation improves on all tasks, compared to the original embeddings for Glove-wiki, although the results for the NER task are not statistically significant. Gaussian perturbation significantly improves over the original Glove-twitter embeddings for the NER task, and over the word2vec hard debiasing approach on part-of-speech tagging. Overall, the configuration (with $\epsilon = 1.0, \delta = 1e-3$) incurs the least decrease with respect to the original word embeddings. The noiseless approach using quantization provides significantly better results for all tasks compared to the full precision representation.

5.3 Word Embedding t-SNE Visualization

We perform a qualitative analysis similar to Gonen and Goldberg [18] by projecting words into the direction $\overrightarrow{he} - \overrightarrow{she}$. After excluding a list of gender specific words [18], we collect the top 500 male-biased and top 500 female-biased words. Figure 1 shows the t-SNE [46] visualization of these 1000 words for both the Gaussian perturbation and the quantization method. From the fairness perspective,



(a) t-SNE visualization for Gaussian perturbed word embedding.



(b) t-SNE visualization for quantized word embedding.

Figure 1: The t-SNE visualization of 1000 gender biased words. These words become more indistinguishable as the word embedding becomes more fair.

these words should not be distinguishable in the direction of gender. We observe that using the Gaussian perturbation method these sets of words become harder to distinguish, while with the hard debiased method and HSR representation, there is still a clear gap between gendered terms. For the quantization method, the single bit representation displays a higher separation between sets of terms, since every word is represented by a binary vector, this separation boundary can be expected due to the limited representation. Thus, using a two bit representation makes the separation boundary less clear, improving fairness over using one bit. This also aligns with Table 2 where two bit representation was found to be more fair than one bit.

6 DISCUSSION

As part of the discussion of the implications of our work, we first present a comparison of fairness metrics on the original word embeddings, without debiasing, shown in Figure 2. This comparison helps us to establish a baseline to understand how corpus selection and training strategies may influence word embedding fairness. Not all embeddings are created equal, and this affects the bias they display.

Another family of embedding training is contextual embedding models, which can have dynamic interpretation of words given different context. Their training design is different from traditional word embedding algorithms. One classical model is the Bert model [11]. The discussion of debiasing methods for such models is beyond the scope of this paper. For the completeness of discussion,

Table 2: Summary of fairness evaluation metrics for debiased embeddings ("Fairness Measurement") and summary of downstream task performance ("Downstream Impact"), according to configurations described in Table 1. The baseline is in italics, and the best results are in bold. A dagger (†) indicatees the experimental result is statistically significantly better than baseline ($p < 0.05$), and an asterisk (⋆) indicates the experimental result is statistically significantly worse than the baseline.

Word Embedding	Fairness Measurement				Downstream Impact		
	WEAT	WEAT ES	RND	RNSB	POS tagging (accuracy)	NER (f1-score)	chunking (f1-score)
conceptnet	*0.143*	*0.620*	***0.031***	***0.007***	***93.622***	***82.940***	***87.432***
conceptnet-1	**0.055**	**0.283**	0.051	0.082	89.088*	70.246*	76.544*
conceptnet-2	0.073	0.334	0.050	0.094	88.788*	68.784*	75.024*
conceptnet-3	0.084	0.413	0.448	0.134	86.276*	58.298*	68.236*
conceptnet-4	0.056	0.309	0.482	0.149	83.902*	55.718*	65.736*
fasttext-wiki	*0.259*	*0.675*	*0.075*	***0.021***	***96.956***	***88.710***	***90.050***
fasttext-wiki-1	0.078	**0.292**	**0.041**	0.028	95.322*	82.926*	85.238*
fasttext-wiki-2	0.081	0.353	0.043	0.061	94.456*	80.202*	82.730*
fasttext-wiki-3	0.067	0.348	0.108	0.084	92.006*	68.708*	76.434*
fasttext-wiki-4	**0.052**	0.428	0.129	0.094	91.428*	64.660*	74.184*
glove-twitter	*0.318*	*0.706*	*0.257*	*0.137*	***96.002***	*85.024*	***88.284***
glove-twitter-1	0.313	0.714	0.261	0.123	95.968	**85.648**†	87.844
glove-twitter-2	0.300	0.692	0.256	0.123	95.984	85.150	88.210
glove-twitter-3	0.175	0.530	0.224	**0.115**	95.304*	83.154*	85.870*
glove-twitter-4	**0.116**	**0.466**	**0.209**	0.122	94.408*	80.368*	82.834*
glove-wiki	*0.539*	*0.972*	*0.263*	*0.168*	*95.976*	*88.496*	*86.410*
glove-wiki-1	0.519	0.966	0.254	0.151	**96.012**†	**88.662**	86.470
glove-wiki-2	0.518	0.961	0.254	0.167	96.006	88.534	**86.696**†
glove-wiki-3	0.261	0.700	0.183	0.126	95.320*	85.784*	83.742*
glove-wiki-4	**0.185**	**0.573**	0.159	0.146	94.336*	82.712*	81.990*
glove-wiki-ran [24]	0.380	0.819	**0.026**	**0.039**	93.484*	84.436*	83.074*
lexvec	*0.326*	*0.726*	*0.686*	*0.153*	***93.500***	*85.092*	*85.154*
lexvec-1	0.323	0.726	0.682	0.119	93.456	85.180	85.204
lexvec-2	0.321	0.715	0.687	0.119	93.440*	84.942	**85.452**
lexvec-3	0.304	**0.702**	0.684	**0.114**	93.282*	**85.316**	84.776
lexvec-4	**0.301**	0.714	**0.633**	0.131	93.058*	85.166	83.970
word2vec	*0.381*	*0.813*	*0.145*	*0.047*	*91.412*	***87.686***	*84.126*
word2vec-1	0.347	0.773	0.135	0.087	91.284*	87.630	83.794
word2vec-2	0.318	0.753	0.139	0.086	91.272*	87.268	83.290
word2vec-3	**0.107**	0.469	0.070	0.064	88.750*	82.650*	77.506*
word2vec-4	0.108	**0.419**	0.078	0.080	86.942*	76.638*	73.330*
word2vec-hsr [48]	0.487	0.925	0.116	0.029	88.780*	73.682*	81.208*
w2v-hard-debiased [8]	0.153	0.577	**0.022**	**0.014**	**91.920**†	86.868	**84.634**
w2b-wiki	*0.310*	*0.649*	*0.799*	*0.260*	*94.972*	*81.476*	*85.046*
w2b-wiki-1bit	0.250	0.691	**0.203**	0.192	**95.844**†	84.624†	**87.812**†
w2b-wiki-2bits	**0.202**	**0.631**	0.446	**0.176**	95.710†	**84.858**†	87.220†
w2b-marco	***0.122***	*0.500*	*0.882*	*0.203*	*91.336*	*70.460*	*79.062*
w2b-marco-1bit	0.406	0.439	0.095	**0.038**	**92.304**†	**75.506**†	79.448
w2b-marco-2bits	0.243	**0.370**	**0.088**	0.114	92.108†	75.372†	**80.408**†

we show the results of our approach to debiasing global word embeddings generated from Bert [11] in the Appendix. We found that the proposed approaches improve fairness for Bert embeddings, while maintaining performance on downstream tasks.

In more detail, we notice that the results for RND are often different from those of WEAT, even though their computation process is similar. WEAT is computed using cosine similarity, whereas RND is based on Euclidean distance. Although these two methods are positively related when inputs are normalized, their fairness evaluation results are quite different. One important factor is that word embedding vectors are unbounded during training, and they are not normalized after training. This is because the choice of whether

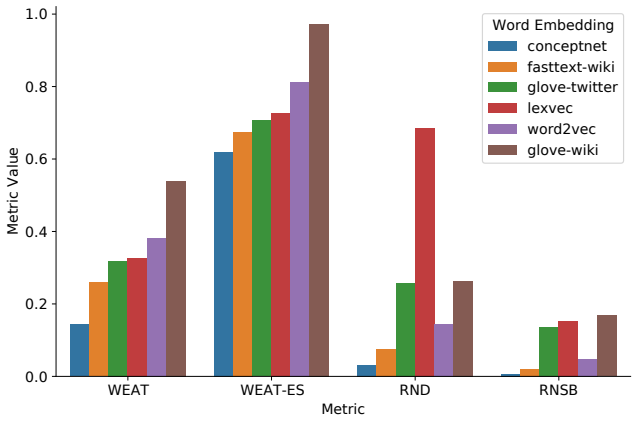


Figure 2: Comparison of fairness metrics across unmodified public pre-trained word embeddings. Values closer to zero imply a more fair word embedding. Fairness of different word embeddings are consistent for WEAT and WEAT-ES.

or not to normalize depends on the application. For example, similarity and word relation tasks often benefit from normalization [47], but word significance tasks need the original vector for the vector length [40]. In addition, unlike WEAT which computes word pair distance, RND calculates the distance between a word embedding vector and the vector average of a list of words. The different results of WEAT and RND can be attributed to word embedding vector length and the different computation targets, thus it is possible that LexVec is more fair than Word2vec under WEAT, but less fair under RND. RNSB, on the other hand, is not similar to any other fairness metrics, since it transforms the fairness problem into a classification problem. RNSB uses the output probability as the relation between the input word and predicted word. A fair word embedding should result in the classifier giving a uniform prediction probability, as it would not be biased toward any predicted word.

We also study the relation between privacy budget and downstream impact. According to Eq. 1 and Eq. 2, a low privacy budget requires that the training output from neighboring corpus stays close to the output from the original corpus. In terms of differential privacy, this results in a large Gaussian perturbation, so that all word embedding vectors are equally random regardless of training corpus. Table 1 shows that small ϵ and δ are associated with large Gaussian noise. For noiseless privacy, this leads to representations with fewer bits, with word embedding vectors becoming very similar due to the smaller number of possible representations. Fairness improves along with Gaussian perturbation, as shown in Proposition 3.1. In some cases, as expected, these improvements can come at the expense of utility in downstream tasks due to the added noise. However, this is not universally the case, since LexVec, Glove and Word2vec embeddings maintained their utility, even when adding Gaussian noise. Further, in many cases, depending on the application, the loss in accuracy might not even be noticeable in practice.

Quantization generally improves fairness with no loss in utility regardless of the privacy budget. This can be attributed to the fact that quantization is done as part of training process, and the optimization process is equivalent to a structured regularization [3]. Therefore, a well trained quantized embedding will show greater improvements than post-processing a trained embedding with Gaussian perturbation.

7 CONCLUSION

In this work, we proposed privacy inspired methods to improve word embedding fairness. The advantage of our approach is that the methods are unsupervised as they do not require prior identification of encoded biases, or predefined lists of biased terms. We investigated two different methods borrowed from (noise-based) differential privacy and noiseless privacy. Gaussian perturbation as a classical differential privacy algorithm can be used as a post-processing method to improve word embedding fairness, making it intuitive and efficient, in situations where the downstream task accuracy is less critical. Quantization as a noiseless privacy algorithm is applied to training the word embedding with a low bit representation. The gains in fairness come at the cost of efficiency at training time. However, the resulting embeddings are smaller. Our experiments show that both methods improve word embedding fairness while maintaining the word embedding effectiveness on downstream tasks, with the greatest gains from the noiseless approach, which improves fairness as well as downstream tasks.

APPENDIX

Word Embeddings derived from Contextual Embedding Models. Following Hollenstein et al. [22], we generate global word embeddings from Bert by feeding each single word as a sentence, and we represent each word using the average of outputs from the second layer to the last layer. Table 3 lists the details about the configuration for the word embeddings derived from Bert, and Table 4 presents related experimental results.

Table 3: Bert-based global word embedding privacy configuration.

Word Embedding	Dim.	Configuration
bert-1	768	$\epsilon = 1.00, \delta = 1e-03, N(0, 0.06)$
bert-2	768	$\epsilon = 1.00, \delta = 1e-05, N(0, 0.09)$
bert-3	768	$\epsilon = 0.10, \delta = 1e-03, N(0, 0.40)$
bert-4	768	$\epsilon = 0.10, \delta = 1e-05, N(0, 0.71)$

Word Embedding Sensitivity Heuristic. For differential privacy the larger ϵ stands for larger privacy budget, since Eq. 1 will allow for more distortion between neighboring datasets. The noise amount depends on privacy budget (ϵ, δ) and also the sensitivity of the algorithm output. In this work, we also provide a simple estimation for word embedding based on the observation on the change in pointwise mutual information (PMI).

PMI is calculated as follows: Let $\mathbf{W} \in \mathbb{R}^{n \times d}$ be a word embedding matrix, and there are n words in total. Each word is represented in an embedding vector $\mathbf{w}_i \in \mathbb{R}^d$ for $i \in \{1, \ldots, n\}$. Note that word embedding vectors are trained results from given corpus C. Let context vector $\mathbf{c}_i$ for word $\mathbf{w}_i$ be the average of its neighboring

Table 4: Experimental results on Bert-based global word embedding.

Word Embedding	Fairness Measurement				Downstream Impact		
	WEAT	WEAT ES	RND	RNSB	POS tagging (accuracy)	NER (f1-score)	chunking (f1-score)
bert	***0.027***	*0.535*	*0.213*	*0.085*	*93.234*	*81.634*	*87.378*
bert-1	0.030	0.547	0.195	0.090	**93.264**	**82.254**	**87.542**
bert-2	0.028	0.542	0.190	0.124	93.204	81.538	87.358
bert-3	0.030	0.436	**0.138**	0.096	91.756*	78.694*	83.648*
bert-4	0.034	**0.298**	0.091	**0.082**	90.210*	75.236*	80.482*

words within certain window size L, i.e., $\mathbf{c}_i = \frac{1}{2L}\sum_j \mathbf{w}_j$ where $j \in \{i-L,\dots,i+L\}\backslash\{i\}$. The skip-gram training algorithm [26] with negative sampling value one has been shown that

$$\mathbf{w}_i^T \mathbf{c}_i = \log \frac{\#(w,c)|C|}{\#(w)\#(c)} = \mathrm{PMI}(\mathbf{w}_i, \mathbf{c}_i), \tag{8}$$

where $\#(\cdot)$ gives number of occurrences of word, context or word-context pairs in training corpus C, and $|C|$ gives the total number of words. This output value is the PMI between a word and its context.

According to Eq. 8, we show the PMI change between the neighboring corpus which is defined as differing in a single word.

$$\begin{aligned}
\Delta(PMI) &= \log \frac{\#(w,c)|C|}{\#(w)\#(c)} - \log \frac{(\#(w,c)-1)(|C|-1)}{(\#(w)-1)\#(c)} \\
&= \log \frac{|C|\#(w,c)(\#(w)-1)}{(|C|-1)(\#(w,c)-1)\#(w)} \\
&\approx \log \frac{\#(w,c)(\#(w)-1)}{(\#(w,c)-1)\#(w)} \quad (since\ |C| \gg \#(w) \geq \#(w,c)) \\
&< \log \frac{\#(w,c)}{(\#(w,c)-1)} \leq \log 2.
\end{aligned} \tag{9}$$

Let $\Delta(\mathcal{A})$ be the algorithm sensitivity. It represents the maximum distance of word embedding vectors trained from neighboring corpus. We show the relation between $\Delta(\mathcal{A})$ and $\Delta(PMI)$ as following:

$$\begin{aligned}
\Delta(PMI) &= ||\mathbf{w}_i^T\mathbf{c}_i - \mathbf{w}_i'^T\mathbf{c}_i + \mathbf{w}_i'^T\mathbf{c}_i - \mathbf{w}_i'^T\mathbf{c}_i'|| \\
&= ||(\mathbf{w}_i - \mathbf{w}_i')^T\mathbf{c}_i + \frac{1}{2L}\sum_{j=1}^{2L}(\mathbf{w}_j - \mathbf{w}_j')^T\mathbf{w}_i'|| \\
&= ||\frac{1}{2L}\sum_{j=1}^{2L}(\mathbf{w}_i - \mathbf{w}_i')^T\mathbf{w}_j + \frac{1}{2L}\sum_{j=1}^{2L}(\mathbf{w}_j - \mathbf{w}_j')^T\mathbf{w}_i'|| \\
&< \frac{1}{2L}\sum_{j=1}^{2L}\Delta(\mathcal{A})||\mathbf{w}_j|| + \frac{1}{2L}\sum_{j=1}^{2L}\Delta(\mathcal{A})||\mathbf{w}_i'|| < 2\Delta(\mathcal{A})M,
\end{aligned} \tag{10}$$

where $M = \max_i\{||\mathbf{w}_i||, ||\mathbf{w}_i'||\}$. In practice, we set $M \approx \max_i ||\mathbf{w}_i||$ since it's hard to compute $\mathbf{w}_i'$ for all neighboring corpus. Given that $\Delta(PMI) < \log 2$, we simply assume that $2\Delta(\mathcal{A})M \approx \log 2$, so we use $\Delta(\mathcal{A}) = \log 2/2M$ as the heuristic in our experiments.

Fairness Improvement with Gaussian Perturbation.

Lemma .1. *Let $v_1, v_2 \in \mathbb{R}^n$ be two unit vectors and θ be the angle between them. Then for $r \in \mathbb{R}^+$,*

$$\frac{1}{V_n}\int_{S_n} \cos(v_1, v_2 + rx)dx = f(r)\cos(v_1, v_2) \tag{11}$$

where $0 < f(r) < 1$. Here V_n is the surface area of the n-dimensional unit sphere S_n.

Proof. Without loss of generality, we assume v_1, v_2 are such that $v_1 = (0,\dots,0,1,0)$ and $v_2 = (0,\dots,0,\cos\theta,\sin\theta)$. With spherical coordinate, we adopt the classical notation for n-dimensional spherical coordinates, where $x = (x_1,\dots,x_n)$ with

$$\begin{aligned}
x_1 &= \cos\phi_1, \\
x_i &= \sin\phi_1\cdots\sin\phi_{i-1}\cos\phi_i \quad 1 < i < n, \\
x_n &= \sin\phi_1\cdots\sin\phi_{n-2}\sin\phi_{n-1}.
\end{aligned}$$

Let $J_n = \sin^{n-2}\phi_1\cdots\sin\phi_{n-2}$, $\gamma = \sin\phi_1\cdots\sin\phi_{n-2}$ and $d\hat{\Phi} = d\phi_1\dots d\phi_{n-2}$. Then we have

$$\begin{aligned}
&\int_{S_n}\cos(v1, v2 + rx)dx \\
&= \int_0^\pi\cdots\int_0^\pi\int_0^{2\pi}\frac{\cos\theta + r\gamma\cos\phi_{n-1}}{\sqrt{1 + r^2\gamma^2 + (1-\gamma^2) + 2r\gamma\cos(\phi_{n-1}-\theta)}}J_n d\phi_{n-1}d\hat{\Phi} \\
&= \int_0^\pi\cdots\int_0^\pi\int_0^{2\pi}\frac{\cos\theta + r\gamma\cos(\phi+\theta)}{\sqrt{1 + r^2\gamma^2 + (1-\gamma^2) + 2r\gamma\cos\phi}}J_n d\phi d\hat{\Phi} \\
&= \int_0^\pi\cdots\int_0^\pi\int_0^{\pi}\frac{2\cos\theta + r\gamma(\cos(\phi+\theta) + \cos(-\phi+\theta))}{\sqrt{1 + r^2\gamma^2 + (1-\gamma^2) + 2r\gamma\cos\phi}}J_n d\phi d\hat{\Phi} \\
&= \cos\theta\int_0^\pi\cdots\int_0^\pi\int_0^{\pi}\frac{2 + 2r\gamma\cos\phi}{\sqrt{1 + r^2\gamma^2 + (1-\gamma^2) + 2r\gamma\cos\phi}}d\phi J_n d\hat{\Phi}
\end{aligned}$$

Let $g(r,\phi_1,\dots,\phi_{n-2}) = \int_0^\pi \frac{2+2r\gamma\cos\phi}{\sqrt{1+r^2\gamma^2+(1-\gamma^2)+2r\gamma\cos\phi}}d\phi$. $g(\cdot)$ can be explicitly represented by complete elliptic integrals of first and second kind for fixed $r,\phi_1,\dots,\phi_{n-2}$ and $0 < g(r,\phi_1,\dots,\phi_{n-2}) < 2\pi$. Thus we have

$$\begin{aligned}
&\frac{1}{V_n}\int_{S_n}\cos(v1, v2 + rx)dx \\
=&\frac{1}{V_n}\cos\theta\int_0^\pi\cdots\int_0^\pi g(r,\phi_1,\dots,\phi_{n-2})J_n d\hat{\Phi} = f(r)\cos\theta
\end{aligned}$$

and

$$0 < f(r) = \frac{1}{V_n}\int_0^\pi\cdots\int_0^\pi g(r,\phi_1,\dots,\phi_{n-2})J_n d\hat{\Phi} < 1$$

□

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Pablo Badilla, Felipe Bravo-Marquez, and Jorge Pérez. 2020. WEFE: The Word Embeddings Fairness Evaluation Framework. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 430–436. https://doi.org/10.24963/ijcai.2020/60 Main track.

[3] Yu Bai, Yu-Xiang Wang, and Edo Liberty. 2019. ProxQuant: Quantized Neural Networks via Proximal Operators. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=HyzMyhCcK7

[4] Borja Balle and Yu-Xiang Wang. 2018. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*. PMLR, 394–403.

[5] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research* 3 (2003), 1137–1155.

[6] Raghav Bhaskar, Abhishek Bhowmick, Vipul Goyal, Srivatsan Laxman, and Abhradeep Thakurta. 2011. Noiseless database privacy. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 215–232.

[7] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (Technology) is Power: A Critical Survey of "Bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 5454–5476. https://doi.org/10.18653/v1/2020.acl-main.485

[8] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems* 29 (2016), 4349–4357.

[9] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.

[10] Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Association for Computational Linguistics, Berlin, Germany, 1–6. https://doi.org/10.18653/v1/W16-2501

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[12] Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)*. 1383–1392.

[13] Yitao Duan. 2009. Privacy without noise. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 1517–1520.

[14] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407.

[15] Farhad Farokhi. 2021. Noiseless Privacy: Definition, Guarantees, and Applications. *IEEE Transactions on Big Data* (2021).

[16] John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis* (1957).

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

[18] 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 NAACL-HLT*.

[19] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[20] Anthony G Greenwald, Debbie E McGhee, and Jordan LK Schwartz. 1998. Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology* 74, 6 (1998), 1464.

[21] Krzysztof Grining and Marek Klonowski. 2017. Towards extending noiseless privacy: Dependent data and more practical approach. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 546–560.

[22] Nora Hollenstein, Adrian van der Lek, and Ce Zhang. 2020. CogniVal in Action: An Interface for Customizable Cognitive Word Embedding Evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*. International Committee on Computational Linguistics (ICCL), Barcelona, Spain (Online), 34–40. https://doi.org/10.18653/v1/2020.coling-demos.7

[23] Christoph Hube, Maximilian Idahl, and Besnik Fetahu. 2020. Debiasing word embeddings from sentiment associations in names. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 259–267.

[24] Vaibhav Kumar, Tenzin Singhay Bhotia, and Tanmoy Chakraborty. 2020. Nurse is closer to woman than surgeon? mitigating gender-biased proximities in word embeddings. *Transactions of the Association for Computational Linguistics* 8 (2020), 486–503.

[25] Maximilian Lam. 2018. Word2bits-quantized word vectors. *arXiv preprint arXiv:1803.05651* (2018).

[26] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems* 27 (2014), 2177–2185.

[27] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. 2021. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing* 461 (2021), 370–403.

[28] Xuebo Liu, Derek F Wong, Yang Liu, Lidia S Chao, Tong Xiao, and Jingbo Zhu. 2019. Shared-private bilingual word embeddings for neural machine translation. *arXiv preprint arXiv:1906.03100* (2019).

[29] Thomas Manzini, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. Black is to Criminal as Caucasian is to Police: Detecting and Removing Multiclass Bias in 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 615–621. https://doi.org/10.18653/v1/N19-1062

[30] Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Using Large Corpora* (1994), 273.

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

[32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[33] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *choice* 2640 (2016), 660.

[34] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[35] Dwaipayan Roy, Debasis Ganguly, Sumit Bhatia, Srikanta Bedathur, and Mandar Mitra. 2018. Using word embeddings for information retrieval: How collection and term normalization choices affect performance. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 1835–1838.

[36] Alexandre Salle, Aline Villavicencio, and Marco Idiart. 2016. Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, 419–424. https://doi.org/10.18653/v1/P16-2068

[37] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[38] Erik F Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. *arXiv preprint cs/0009008* (2000).

[39] EF Tjong Kim Sang and F De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003, Edmonton, Canada*. Morgan Kaufman Publishers, 142–145.

[40] Adriaan MJ Schakel and Benjamin J Wilson. 2015. Measuring word significance using distributed representations of words. *arXiv preprint arXiv:1508.02297* (2015).

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

[42] Emeralda Sesari, Max Hort, and Federica Sarro. 2022. An Empirical Study on the Fairness of Pre-trained Word Embeddings. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*. Association for Computational Linguistics, Seattle, Washington, 129–144. https://doi.org/10.18653/v1/2022.gebnlp-1.15

[43] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. 4444–4451. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972

[44] Roger Alan Stein, Patricia A Jaques, and Joao Francisco Valiati. 2019. An analysis of hierarchical text classification using word embeddings. *Information Sciences* 471 (2019), 216–232.

[45] Chris Sweeney and Maryam Najafian. 2019. A transparent framework for evaluating unintended demographic bias in word embeddings. In *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics. 1662–1667.

[46] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[47] Benjamin J Wilson and Adriaan MJ Schakel. 2015. Controlled experiments for word embeddings. *arXiv preprint arXiv:1510.02675* (2015).

[48] Zekun Yang and Tianlin Liu. 2020. Causally denoise word embeddings using half-sibling regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9426–9433.

[49] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 15–20. https://doi.org/10.18653/v1/N18-2003