A Neighbourhood-Aware Differential Privacy Mechanism for Static Word Embeddings

Danushka Bollegala[1,2*] **Shuichi Otake**[3] **Tomoya Machide**[4] **Ken-ichi Kawarabayashi**[3]
University of Liverpool[1], Amazon[2], National Institute of Informatics[3]
International Professional University of Technology in Tokyo[4]

Abstract

We propose a Neighbourhood-Aware Differential Privacy (NADP) mechanism considering the neighbourhood of a word in a pre-trained static word embedding space to determine the minimal amount of noise required to guarantee a specified privacy level. We first construct a nearest neighbour graph over the words using their embeddings, and factorise it into a set of connected components (i.e. neighbourhoods). We then separately apply different levels of Gaussian noise to the words in each neighbourhood, determined by the set of words in that neighbourhood. Experiments show that our proposed NADP mechanism consistently outperforms multiple previously proposed DP mechanisms such as Laplacian, Gaussian, and Mahalanobis in multiple downstream tasks, while guaranteeing higher levels of privacy.

1 Introduction

Increasingly more NLP models have been trained on private data such as medical conversations, social media posts and personal emails (Abdalla et al., 2020; Lyu et al., 2020b; Song and Shmatikov, 2019). However, we must ensure that sensitive information related to user privacy is not leaked during any stage of the model training process. To protect user privacy, Differential Privacy (DP) mechanisms add random noise to the training data (Feyisetan and Kasiviswanathan, 2021; Krishna et al., 2021; Feyisetan et al., 2020). However, it remains a challenging task to balance the trade-off between user **privacy** vs. **performance** in downstream NLP tasks.

We propose **Neighbourhood-Aware Differential Privacy** (NADP) mechanism, which consists of three steps. First, given a set of words, we compute a nearest neighbour graph considering the similarity between the words (represented by the vertices of the nearest neighbour graph) computed using their word embeddings. Second, we compute the connected components in the nearest neighbour graph to find the *neighbourhoods* of words. Third, we apply Gaussian noise to all words in each neighbourhood, such that the variance of the noise is determined by the words in that neighbourhood.

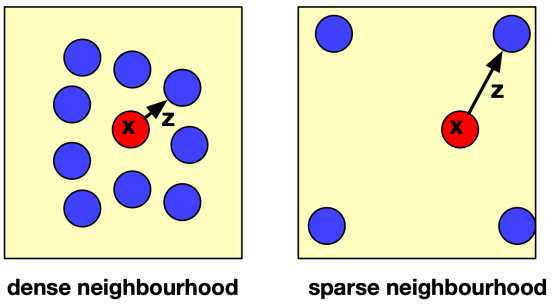


Figure 1: Anonymizing a target word (shown in red) in a dense (left) vs. a sparse (right) neighbourhoods of words (shown in blue). In the sparse neighbourhood, NADP adds a higher level of perturbation noise z to the target word embedding x in order to protect its privacy by disguising it among its neighbours, while in a dense neighbourhood it adds less noise.

As illustrated in Figure 1, if all words in a neighbourhood are highly similar to each other (i.e. a *dense* neighbourhood), it would require less perturbation noise to anonymise a word because the addition of small noise can easily *hide* the corresponding word embedding among its neighbours. On the other hand, if the words in a neighbourhood are not very similar to each other (i.e. a *sparse* neighbourhood) we must add higher levels of perturbation noise to a word embedding because its nearest neighbour would be further away in the embedding space. Because words in a language is a discrete set (unlike images for example), there does not exist a word corresponding to all points in the embedding space. Therefore, if we do not add sufficient amount of noise in a sparse

*Contact author: danushka@liverpool.ac.uk

neighbourhood, we run the risk of easily discovering the target word via a simple nearest neighbour search. Instead of adding the same level of noise to all words in a vocabulary as done in prior DP mechanisms, NADP attempts to minimise the total amount of noise by assigning low noise in dense neighbourhoods and high noise in sparse neighbourhoods. NADP has provable DP guarantees as shown by our theoretical analysis. Moreover, NADP has the following desirable properties that makes it attractive when used for NLP tasks.

(a) In NADP, **noise vectors are sampled from the Gaussian distribution**. Many static word embedding algorithms (Pennington et al., 2014; Arora et al., 2016; Mikolov et al., 2013) learn embeddings in the ℓ_2 space. Moreover, the squared ℓ_2 norm of a word embedding is known to positively correlate with the frequency of the word in the training corpus (Arora et al., 2016), while the joint co-occurrence probability of a set of words positively correlates with the squared ℓ_2 norm of the sum of the corresponding word embeddings (Bollegala et al., 2018). Therefore, it is natural to consider Gaussian noise, which corresponds to the ℓ_2 embedding space used by many static word embedding learning methods rather than the more widely-used Laplacian noise, which relates to the ℓ_1 norm.

(b) Unlike previously proposed DP mechanisms for word embeddings (Feyisetan et al., 2020; Feyisetan and Kasiviswanathan, 2021; Krishna et al., 2021; Xu et al., 2020), NADP **dynamically adjusts the level of noise added to a word embedding considering its neighbourhood**. This enables us to optimally allocate a fixed noise budget over a vocabulary.

(c) NADP **adds noise directly to the word embeddings** and does not perform decoding after the noise addition step (Krishna et al., 2021). Decoding is a deterministic process and does not affect DP. Many NLP applications such as text classification, clustering etc. require the input text to be represented in some vector space, and we can use the noise-added input text representations straightaway in such applications without requiring to first decode it back to text. In situations where users train word embeddings on private data on their own and only send/release the embeddings to external machine learning services, we only need to anonymise the word embeddings (Feyisetan and Kasiviswanathan, 2021).

Results: *Utility experiments* (§ 5.1) conducted over four downstream NLP tasks show that NADP consistently outperforms previously proposed Laplacian, Gaussian and Mahalanobis mechanisms in downstream tasks. We conduct *privacy experiments* (§ 5.2) to evaluate the level of privacy guaranteed by a DP mechanism for word embeddings. Specifically, we estimate the probability of correctly predicting a word from its perturbed word embedding using the overlap between nearest neighbour sets. To evaluate the level of privacy protected for the entire set of word embeddings, we compute the skewness of the distribution of prediction probabilities. We find that NADP reports near-zero skewness values across a broad range of privacy levels, ϵ, which indicates significantly stronger privacy guarantees compared to other DP mechanisms. Source code implementation of our NADP is publicly available.[1]

2 Related Work

Learning models from data with DP guarantees has been studied under *private learning* (Kasiviswanathan et al., 2008). Abadi et al. (2016) proposed a DP stochastic gradient descent by adding Gaussian noise to the gradient of the loss function. Rogers et al. (2016) combined multiple DP algorithms using adaptive parameters. However, compared to continuous input spaces such as in computer vision (Zhu et al., 2020), DP mechanisms for the discrete inputs such as text remain understudied.

Wang et al. (2021) proposed *WordDP* to achieve certified robustness against word substitution attacks in text classification. However, WordDP does *not* seek DP protection for the training data as we consider here, and uses DP randomness for certified robustness during inference time with respect to a testing input. Krishna et al. (2021) proposed AdePT, an autoencoder-based approach to generate differentially private text transformations. However, Habernal (2021) showed that AdePT is *not* differentially private as claimed and proved weaker privacy guarantees. DPText (Alnasser et al., 2021; Beigi et al., 2019) uses an autoencoder to obtain a text representation and adds Laplacian noise to create private representations. However, Habernal (2022) proved that the use of reparametrisation trick for the inverse continuous

[1] https://github.com/shuichiotake/NADP

density function in DPText is inaccurate and that DPText violates the DP guarantees. Such prior attempts show the difficulty in developing theoretically correct DP mechanisms for NLP.

Lyu et al. (2020a) proposed DP Neural Representation (DPNR) to preserve the privacy of text representations by first randomly masking words from the input texts and then adding Laplacian noise. However, unlike NADP DPNR uses a neighbourhood insensitive fixed Laplacian noise distribution. Feyisetan et al. (2020) proposed a DP mechanism where they first add Laplacian noise to word embeddings and then return the nearest neighbour to the noise-added embedding as the output. However, the ℓ_2 norm of the noise vector scales almost linearly with the dimensionality of the embedding space. To address this issue, in their subsequent work (Feyisetan and Kasiviswanathan, 2021), they projected the word embeddings to a lower-dimensional space before adding Laplacian noise.

Xu et al. (2020) proposed Mahalanobis DP mechanism, which adds elliptical noise considering the covariance structure in the embedding space. Unlike the Gaussian or Laplacian mechanisms, Mahalanobis mechanism adds heterogeneous noise along different directions such that words in sparse regions in the embedding space have sufficient likelihood of replacement without sacrificing the overall utility. They show that Mahalanobis mechanism to be superior to Laplacian mechanism. Mahalanobis mechanism is a special instance of metric (Lipschitz) DP originated in privacy-preserving geolocation studies (Andrés et al., 2013), where Euclidean distance was used as the distance metric. Although metric DP considers the distance between two data points, it does not consider all of the nearest neighbours for each data point when deciding the level of noise that must be applied to a particular data point, unlike our NADP mechanism. Li et al. (2018) used adversarial learning to build NLP models such as part-of-speech (PoS) taggers that cannot predict the writer's age or sex, while can accurately predict the PoS tags. Despite their empirical success, this approach does not have any formal DP guarantees. In contrast, our focus is provably DP mechanisms with formal guarantees.

All of the prior work described thus far, except DPNR and AdePT, focus on static word embeddings as we do in this paper. A natural future extension of this work is DP mechanisms for the contextualised embeddings. However, computational and practical properties of static word embeddings such as, being lightweight to both compute and store, are attractive for resource (e.g. GPU and RAM) limited mobile devices. Considering that such personal mobile devices are used by billions of users and contain highly private data, DP mechanisms for static word embeddings remains an important research topic. Moreover, Gupta and Jaggi (2021) showed that it is possible to distil static word embeddings from pretrained language models that have comparable performance to contextualised word embeddings.

3 DP for Word Embeddings

Let us denote the d-dimensional embedding of a word x in a vocabulary $\mathcal{X}$ by a vector $\boldsymbol{x} \in \mathbb{R}^d$. We can consider a word embedding algorithm as a function $f : \mathbb{X} \rightarrow \mathbb{R}^d$ that maps the words in a discrete vocabulary space $\mathbb{X}$ to a d-dimensional continuous space $\mathbb{R}^d$. We can use a distance metric, Γ, defined in the embedding space to measure the distance $\Gamma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between two words x_i and x_j such as the Euclidean distance. We can then find the set of top-m nearest neighbours, $\mathcal{S}_m(x)$, from $\mathcal{X}$ using Γ such that for any $y \in \mathcal{S}_m(x)$ and $y' \notin \mathcal{S}_m(x)$, $\Gamma(x, y) \leq \Gamma(x, y')$ holds. The Jaccard similarity, $\text{Jaccard}(x, y)$, between two words x and y is defined using their neighbourhoods as in (1).

$$\text{Jaccard}(x, y) = \frac{|\mathcal{S}_m(x) \cap \mathcal{S}_m(y)|}{|\mathcal{S}_m(x) \cup \mathcal{S}_m(y)|} \quad (1)$$

We define two words $x, y \in \mathcal{X}$ to be in a symmetric *neighbouring* relation, $x \simeq y$, if the following two conditions are jointly satisfied:

(a) $x \in \mathcal{S}_m(y)$ or $y \in \mathcal{S}_m(x)$, and

(b) $\text{Jaccard}(x, y) \geq \tau$ for a given threshold $\tau \in [0, 1]$.

One could use conjunction instead of disjuction in condition (a) to enforce a mutual nearest neighbour relation. However, doing so results in a large number of small isolated neighbourhoods because two words might not be *mutual* nearest neighbours unless they are synonyms (or highly related). Relaxing the condition (a) to a disjunction would form neighbourhoods where one word might be a neighbour of another but not the inverse such as in hypernym-hyponym pairs. For

example, *colour* could be a top nearest neighbour of *crimson*, but *crimson* might not be a top nearest neighbour of *colour*, because there are other prototypical colours such as *red*, *green*, *blue*, etc. than *crimson*.

Let us formally define DP for word embeddings. Because each word is assigned a vector by the word embedding learning algorithm, we can add noise to the embedding vectors to *disguise* a word among its nearest neighbours in the embedding space. However, in doing so we will be perturbing the semantics in the embeddings, thus potentially hurting downstream task performance. Therefore, there exists a *trade-off* between the amount of privacy that can be guaranteed by adding random noise to the embeddings vs. the performance of a downstream NLP task that use those embeddings. A random mechanism operating on word embeddings is said to be DP if Definition 1 holds.

Definition 1 (Differential Privacy)**.** *A random mechanism M that takes in a vector in the embedding space $\mathbb{X}$ and maps into a space $\mathbb{Y}$ (i.e. $M : \mathbb{X} \rightarrow \mathbb{Y}$) is (ϵ, δ)-DP with $\epsilon \geq 0$ and $\delta \in [0, 1]$, if for every pair of neighbouring inputs $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and every possible measurable output set $\mathcal{T} \in \mathcal{Y}$ the relationship given by (2) holds:*

$$\Pr[M(\boldsymbol{x}) \in \mathcal{T}] \leq \exp(\epsilon)\Pr[M(\boldsymbol{x}') \in \mathcal{T}] + \delta \tag{2}$$

Here, ϵ represents the level of privacy ensured by M and smaller ϵ values result in stronger privacy guarantees. The global ℓ_2 sensitivity of the embedding space is defined as $\Delta = \sup_{x,x' \in \mathcal{X}, x \simeq x'} ||\boldsymbol{x} - \boldsymbol{x}'||$. Given a set of word embeddings, Δ can be estimated empirically by computing the maximum Euclidean distance between a word x and its most distant neighbour x' in $\mathcal{S}_m(x)$. As an extreme case, let us consider the smallest possible neighbourhood size corresponding to $m = 2$. Estimating Δ in this case would amount to finding the maximum Euclidean distance between any pair of neighboring words $x, x' \in \mathcal{V}$. Moreover, the Δ estimated for $m = 2$ will be larger than the Δ estimated for any other $m(> 2)$ neighbourhood sizes. Therefore, Δ is independent of m and can be estimated via a deterministic process (i.e. measuring all pairwise Euclidean distances) from a given set of word embeddings.

Algorithm 1: Nearest Neighbour Graph Construction

Inputs: Word embeddings $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, top-$m$ for selecting neighbours, and similarity threshold $\tau \in [0, 1]$.
Outputs: Nearest neighbour graph $G(\mathcal{V}, \mathcal{E})$
Initialise $\mathcal{V} = \{x_1, \ldots, x_n\}$, $\mathcal{E} = \{\}$
for $i = 1, \ldots, n$ **do**
  for $x_j \in \mathcal{S}_m(x_i)$ **do**
    if $\text{Jaccard}(x_i, x_j) \geq \tau$ **then**
      $\mathcal{E} = \mathcal{E} + \{(i, j)\}$
Return $G(\mathcal{V}, \mathcal{E})$

3.1 Gaussian Mechanism

Gaussian mechanism uses ℓ_2 norm for estimating the sensitivity due to perturbation and is a more natural fit for word embeddings than, for example, the Laplace mechanism, which is associated with the ℓ_1 norm. Therefore, we use the Gaussian mechanism as the basis for our proposal.

Let us consider a multivariate zero-mean isotropic Gaussian noise distribution, $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$, where $\mathbf{I}_d$ is the unit matrix in the d-dimensional real space and σ is the standard deviation. For each word, $x \in \mathcal{X}$, we sample a random vector $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I}_d)$ and create a noise-added embedding $M_g(\boldsymbol{x})$ for x as given by (3).

$$M_g(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{z} \tag{3}$$

This Gaussian mechanism uses the same σ for all words in the vocabulary and is (ϵ, δ)-DP as claimed in in Theorem 1.[2]

Theorem 1. *For any $\epsilon, \delta \in (0, 1)$, the Gaussian mechanism with $\sigma = \Delta\sqrt{2\log(1.25/\delta)}/\epsilon$ is (ϵ, δ)-DP.*

The proof of Theorem 1 can be found in the Appendix A in (Dwork and Roth, 2014).

4 Neighbourhood-Aware Differential Privacy (NADP)

Our proposed DP-mechanism, Neighbourhood-Aware Differential Privacy (NADP), consists of

[2] As a direct extension, we could set a different standard deviations for each dimension of the embedding space. However, doing so did not result in significant performance gains in our preliminary investigations despite the increased parameters.

three main steps. First, we create a nearest neighbour graph where vertices represent the words as described in § 4.1. Next, we factorise this nearest neighbour graph into a set of mutually exclusive neighbourhoods by finding its connected components as described in § 4.2. Finally, for the words that belong to each connected component, we add random noise sampled from Gaussian distributions with zero mean and *different* standard deviations, determined according to the neighbourhood associated with the corresponding connected component. We prove that the proposed NADP mechanism is DP in § 4.3.

4.1 Nearest Neighbour Graph Construction

To represent the nearest neighbours of a set $\mathcal{X}$ of words, we construct a nearest neighbour graph, $G = G(\mathcal{X}, \simeq)$ with the symmetric neighbouring relation $\simeq$, vertex set $\mathcal{V}(G) = \mathcal{X}$ and edge set $\mathcal{E}(G) = \{(x, x') \mid x \simeq x'\}$. Given the one-to-one mapping between words and the vertices in the graph, for notational simplicity we denote the i-th vertex of the graph by $x_i (\in \mathcal{X})$. Two vertices x_i and x_j are connected by an edge $e_{ij} (\in \mathcal{E})$, if $x_i \simeq x_j$ holds between the corresponding words x_i and x_j. As already explained in § 3, we define two words $x_i, x_j \in \mathcal{X}$ to be in a symmetric neighbouring relation, $x_i \simeq x_j$ if the following two conditions are jointly satisfied: (a) $x_i \in \mathcal{S}_m(x_j)$ or $x_j \in \mathcal{S}_m(x_i)$, and (b) $\text{Jaccard}(x_i, x_j) \geq \tau$ for a predefined similarity threshold $\tau \in [0, 1]$.

The pseudo code for constructing the nearest neighbour graph is shown in Algorithm 1. In our experiments, we set $m = 2$, which considers only the top-2 neighbours (i.e. $\mathcal{S}_2$) to ensure only the highly similar neighbours are connected by edges in the nearest neighbour graph. τ can be used to remove neighbours that have less similarity to a target word across the graph. For example, by setting $\tau = 0.8$, we can ensure that no two words with neighbourhood similarity (measured using the Jaccard coefficient) less than 0.8 will be connected by an edge in $\mathcal{G}$. We empirically study the effect of varying τ on NADP later in our experiments.

4.2 Finding Connected Components

Once a nearest neighbour graph $\mathcal{G}$ is constructed for $\mathcal{X}$, next we identify the regions of neighbours, which we refer to as the *neighbourhoods*. To consider tightly connected neighbourhoods, we propose to factorise $\mathcal{G}$ into a set of mutually exclusive connected components following the procedure described in Algorithm 2. We start by randomly selecting a word x from $\mathcal{X}$ and creating a neighbourhood $\mathcal{X}_1$ consisting all of x's neighbours. We then remove the words in $\mathcal{X}_1$ from $\mathcal{X}$, and repeat this process until all words in $\mathcal{X}$ are included in some neighbourhood. The procedure described in Algorithm 2 for obtaining connected components from $\mathcal{G}$ is simple, efficient and obtains good DP performance in our experiments. Moreover, it does *not* require the number of neighbourhoods, k, to be specified in advance as it would be the case for many clustering-based approaches for graph partitioning such as spectral clustering (von Luxburg, 2007). There is a possibility of obtaining long chains when computing connected components using Algorithm 2. However, we did not encounter this issue in our experiments. This is because the nearest neighbour relation that is defined in § 3 requires both mutual nearest neighbourhood and high Jaccard similarity to be satisfied, which reduces the likelihood of forming long chains. Exploring alternative methods for factorising a given graph into a set of mutually exclusive connected components is deferred to future work.

Algorithm 2: Finding Connected Components

Inputs: Nearest neighbour graph $G(\mathcal{V}, \mathcal{E})$
Outputs: Connected components $\{\mathcal{X}_1, \ldots, \mathcal{X}_k\}$
Define $k = 0$
Define $\mathcal{H}_k = \{\}$
while $\mathcal{X} \setminus \mathcal{H}_k \neq \emptyset$ **do**
  Choose $x \in \mathcal{X} \setminus \mathcal{H}_k$
  $k = k + 1$
  $\mathcal{X}_k = \{x\}$
  Define $\mathcal{X}' = \{x' | x' \simeq x\} \setminus \mathcal{X}_k$
  $\mathcal{X}_k = \mathcal{X}_k \cup \mathcal{X}'$
  $\mathcal{H}_k = \mathcal{H}_k \cup \mathcal{X}_k$
  while $\mathcal{X}' \neq \emptyset$ **do**
    $\mathcal{X}' = \{x' | x' \simeq x$ for some $x \in \mathcal{X}_k\} \setminus \mathcal{X}_k$
    $\mathcal{X}_k = \mathcal{X}_k \cup \mathcal{X}'$
    $\mathcal{H}_k = \mathcal{H}_k \cup \mathcal{X}_k$
Return $\{\mathcal{X}_1, \ldots, \mathcal{X}_k\}$

4.3 Perturbation of Word Embeddings

In this section, we will first prove that NADP satisfies the DP conditions, and then present an al-

gorithm that can be used to add perturbation noise to the words in each neighbourhood. First note that the trivial relation $x \simeq x$ implies the set $\{||\boldsymbol{x}-\boldsymbol{y}|| \mid x \simeq y\}$ is nonempty and hence we can consider the global L_2 sensitivity, $\Delta = \sup_{x \simeq y} ||\boldsymbol{x}-\boldsymbol{y}||$, for any two neighbouring words x and y in the given set of words $\mathcal{X}$. Balle and Wang (2018) proved Theorem 2 that shows a set of word embeddings can be made differentially private by adding Gaussian noise sampled according to Δ, where $\Phi(t)$ the Cumulative Density Function (CDF) of the standard univariate Gaussian distribution, given by (4).

$$\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-y^2/2} dy. \tag{4}$$

Theorem 2 (Balle and Wang (2018))**.** *Let $f : \mathbb{X} \to \mathbb{R}^d$ be a function with global L_2 sensitivity $\Delta > 0$. For any $\varepsilon \geq 0$ and $\delta \in [0,1]$, the Gaussian output perturbation mechanism $M(x) = \boldsymbol{x} + \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ $(\sigma > 0)$ is (ε, δ)-DP if and only if*

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) \leq \delta. \tag{5}$$

The original proof of Theorem 2 is provided in (Balle and Wang, 2018). However, in § C.1 we provide an alternative proof, which is more concise and can be directly extended to the case of multiple neighbourhoods represented by the connected components in the nearest neighbour graph.

Theorem 3 (see § C.2 for proof) states that NADP satisfies DP conditions.

Theorem 3 (main)**.** *Let $\{\mathcal{X}_1, \cdots, \mathcal{X}_k\}$ be the connected components of the graph $G(\mathcal{X}, \simeq)$ and let σ_i $(1 \leq i \leq k)$ be non-negative real numbers such that $\sigma_i > 0$ whenever $\Delta_i = \sup_{x \simeq y, x, y \in \mathcal{X}_i} ||\boldsymbol{x}-\boldsymbol{y}|| > 0$. For any $\boldsymbol{x} \in \mathcal{X}$, let $i(x)$ $(1 \leq i(x) \leq k)$ be the index such that $x \in \mathcal{X}_{i(x)}$. Then, for any $\varepsilon \geq 0$ and $\delta \in [0,1]$, the Gaussian output perturbation mechanism $M(x) = \boldsymbol{x} + \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2_{i(x)} \mathbf{I}_d)$ is (ε, δ)-DP if*

$$\Phi\left(\frac{\Delta_{i(x)}}{2\sigma_{i(x)}} - \frac{\varepsilon\sigma_{i(x)}}{\Delta_{i(x)}}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta_{i(x)}}{2\sigma_{i(x)}} - \frac{\varepsilon\sigma_{i(x)}}{\Delta_{i(x)}}\right) \leq \delta \tag{6}$$

for any $x \in \mathcal{X}$ satisfying $\Delta_{i(x)} > 0$.

Remark 1. *We have $\Delta_{i(x)} = 0$ iff the connected component $\mathcal{X}_{i(x)}$ consists of only one word x.*

Theorem 3 guarantees that the NADP mechanism described in Algorithm 3 for perturbing a set of word embeddings satisfies DP. Specifically, we can first compute u^* globally for all neighbourhoods (Line 4) as the minimiser of $g(u)$ (given by (7)) such that the DP-condition in (5) is satisfied. We can then determine the standard deviation, σ_i, corresponding to each neighbourhood, using u^* and the local sensitivity, Δ_i, computed from that neighbourhood. Finally, we sample noise vectors from $\mathcal{N}(0, \sigma_i^2 \mathbf{I}_d)$ and add to all word embeddings in each $\mathcal{X}_i$.

Algorithm 3: Neighbourhood-Aware Differential Privacy

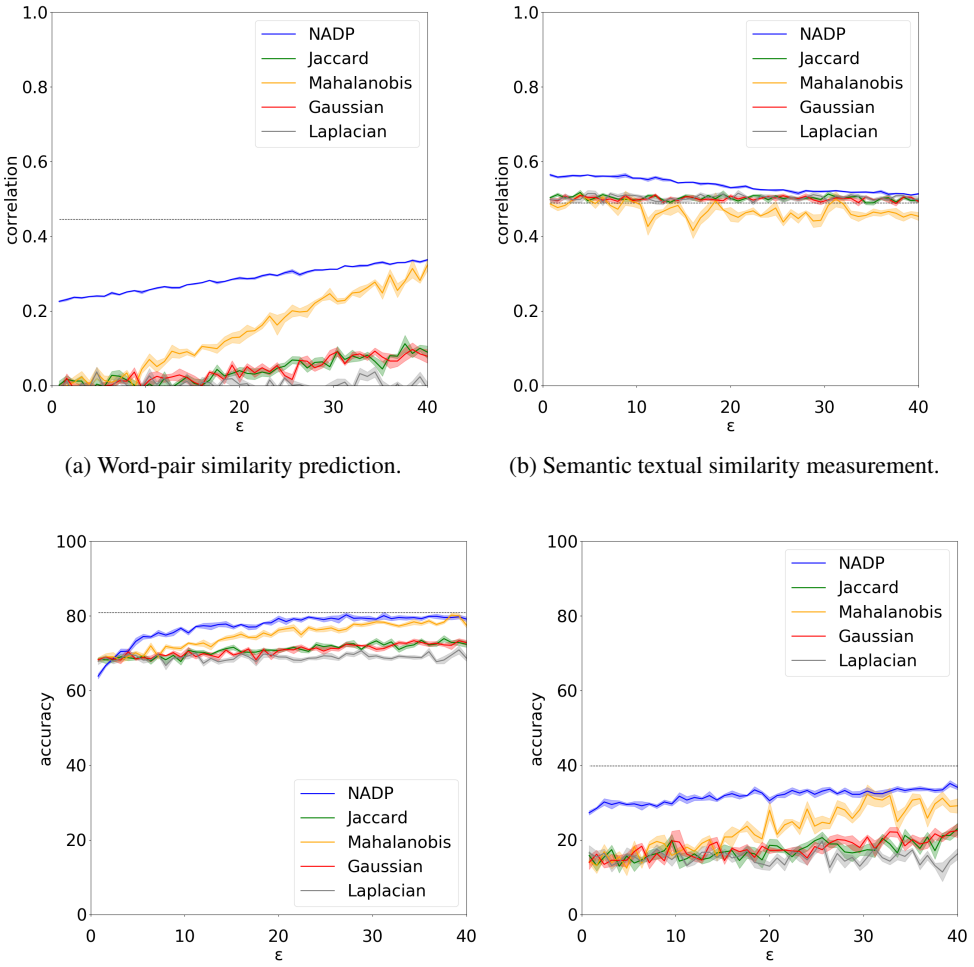
1 **Inputs:** Connected components $\{\mathcal{X}_1, \ldots, \mathcal{X}_k\}$, $\epsilon \geq 0$, $\delta \in [0,1]$
2 **Outputs:** Perturbed word embeddings $\hat{\mathcal{X}} = \{\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_n\}$
3 Define $g(u) = \Phi\left(\frac{1}{2u} - \varepsilon u\right) - e^{\varepsilon}\Phi\left(-\frac{1}{2u} - \varepsilon u\right)$
4 Compute $u^* = \min\{u \in \mathbb{R}_{>0} \mid g(u) \leq \delta\}$
5 Define $\hat{\mathcal{X}} = \{\}$
6 **for** $i = 0, \ldots, k$ **do**
7 Compute $\Delta_i = \sup_{x \simeq y, x, y \in \mathcal{X}_i} ||\boldsymbol{x}-\boldsymbol{y}||$
8 Define $\sigma_i = u^* \Delta_i$
9 **for** $x \in \mathcal{X}_i$ **do**
10 Sample $\boldsymbol{z} \sim \mathcal{N}(0, \sigma_i^2 \mathbf{I}_d)$
11 $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{z}$
12 $\hat{\mathcal{X}} = \hat{\mathcal{X}} + \{\hat{\boldsymbol{x}}\}$
13 Return $\hat{\mathcal{X}}$

5 Experiments

We use the pretrained[3] 300-dimensional GloVe embeddings (Pennington et al., 2014) for 2.8M words, which have also been used in much prior work (Xu et al., 2020; Feyisetan and Kasiviswanathan, 2021) as the static word embeddings.

We build a nearest neighbour graph using the top-1000 frequent words in the English Wikipedia, which resulted in a 73,404 vertex graph. It takes less than 5 minutes to find all connected components of a graph containing 73,404 words used in the paper. Moreover, this is a task independent pre-processing step. Building the neighbourhood graph in a brute force manner requires 3.5

[3]We used 42B token Common Crawl trained embeddings available at https://nlp.stanford.edu/projects/glove/



(a) Word-pair similarity prediction.

(b) Semantic textual similarity measurement.

(c) Text classification.

(d) Odd-man-out.

Figure 2: Performance on utility experiments (§ 5.1) shown in sub-figures (a)-(d). Accuracy and correlation (with human ratings) not decreasing with high privacy (ϵ) levels (corresponding to stronger noise levels by DP mechanisms) is desirable. Performance obtained without adding any noise is shown by the horizontal dotted lines.

hours, while approximate nearest neighbour methods such as SCANN (Guo et al., 2020) an be used to do the same in less than 1 minute with over 95% recall.

In our experiments, we compare NADP against the following DP mechanisms: **Gaussian** mechanism described in § 3.1, **Laplacian** mechanism, where noise vectors are sampled from the Laplace distribution with zero location parameter and with different values of the ϵ scale parameter, **Mahalanobis** mechanism with the recommended parameter values by Xu et al. (2020) (i.e. the Mahalanobis norm $\lambda = 1$ and $\epsilon \in (0, 40]$ are used), which is the current SoTA DP mechanism for static word embeddings.

All of the above mentioned DP-mechanisms apply the same level of random noise to all word embeddings. Therefore, to understand the importance of assigning different levels of noise to different words, we consider a baseline DP mechanism, which we call the **Jaccard** mechanism. We define the density, $\eta(x)$, of the neighbourhood, $\mathcal{S}_k(x)$, of a word x as the average Euclidean distance between x and its nearest neighbours (i.e. $\eta(x) = \frac{1}{k}\sum_{x' \in \mathcal{S}_k(x)} ||\boldsymbol{x} - \boldsymbol{x'}||$). Next, we categorise words into two density categories: dense ($\mathcal{X}_1 = \{x | x \in \mathcal{X}, \eta(x) < \eta_0\}$) vs. sparse ($\mathcal{X}_2 = \{x | x \in \mathcal{X}, \eta(x) \geq \eta_0\}$), based on a density threshold η_0. Our preliminary experiments showed that splitting into more than two categories did not result in significant performance gains. For a word $x \in \mathcal{X}_i$, we sample a random vector $\boldsymbol{n}(x) \sim \mathcal{N}(\mathbf{0}, \sigma_i \mathbf{I}_d)$, for $i \in \{1, 2\}$ and add to $\boldsymbol{x}$. Jaccard is a DP mechanism (see § C.3 for the

proof). Note that the density threshold is used only by the Jaccard mechanism and is not required by NADP. It is determined automatically such that we get approximately similar numbers of words in the dense and sparse sets.

5.1 Utility Experiments

To evaluate the semantic information preserved in word embeddings, we use the following standard tasks that have been used in much prior work for this purpose (Bollegala, 2022; Bollegala and O'Neill, 2022; Tsvetkov et al., 2015; Faruqui et al., 2015): word similarity measurement, semantic textual similarity (STS), Text Classification, Odd-man-out (Stanovsky and Hopkins, 2018). Due to space limitations, we detail the tasks, datasets and evaluation metrics in Appendix A.

Results: Figure 2 shows the performance obtained on utility experiments with noise-added word embeddings for different values of the privacy parameter ϵ, where we use $\tau = 0.5$. The total set of words used in the datasets for all utility experiments is $n = 73404$. Therefore, we set $\delta = 1/73404 \approx 0.000013623$ in all experiments reported in the paper. We repeat each experiment five times and plot the mean and the standard error. Recall that smaller ϵ values provide stronger DP guarantees. From Figure 2, we see that NADP reports the best performance on all four tasks among the methods compared across all ϵ values. Among the other methods, Mahalanobis performs second best to NADP in word-pair similarity prediction, text classification and odd-man-out, but performs worst in STS. In word-pair similarity prediction, text classification and odd-man-out tasks, we see the performance of NADP as well as the other methods increase with ϵ due to less noise being added to the word embeddings.

The performance in STS is comparatively less affected by ϵ because it is a sentence-level comparison task, which considers all perturbed word embeddings in a sentence, whereas the other three are word-level tasks. We see that Jaccard and Gaussian mechanisms perform similarly in all tasks. This is not surprising given that the Jaccard mechanism is drawing the noise vectors from two independent Gaussian distributions. In particular for high ϵ values, we see that Gaussian outperforms Laplacian in word-pair similarity prediction, text classification and odd-man-out tasks. This result implies that for making word embeddings differentially private, the L_2 sensitivity considered in the Gaussian mechanism is more appropriate than the L_1 sensitivity considered in the Laplacian mechanism.

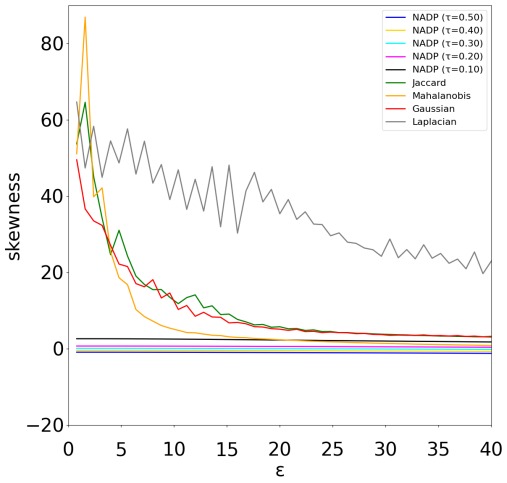


Figure 3: Skewness values for predicting words using their noise-added embeddings. Low skewness values are desirable, and indicate that the prediction probability distribution is similar to the Normal distribution and is not skewed towards a subset of the words.

5.2 Privacy Experiments

To empirically measure the level of privacy protected by a DP mechanism, we consider, $p(x|M(x))$, the probability of predicting the word x using its noise-added embedding $M(x)$, as a metric of privacy provided by a DP mechanism. However, it is difficult to accurately estimate probability densities in discrete spaces due to data sparseness. Therefore, we approximate $p(x|M(x))$ by $\frac{|\mathcal{S}_m(x) \cap \mathcal{S}_m(M(x))|}{|\mathcal{S}_m(x) \cup \mathcal{S}_m(M(x))|}$, using the nearest neighbour sets. It is noteworthy that this is a conservative estimate of $p(x|M(x))$ because, even if all of the nearest neighbours of x and $M(x)$ fully overlap , there will still be a $1/m$ uncertainty ensuring a nonzero level of privacy.

Due to the differences in neighbourhood densities, some words are likely to be influenced more than the others by a DP mechanism. From a DP point of view we are interested in protecting the privacy of all words in the vocabulary and not just for a subset of it. Therefore, to empirically quantify the global effect on privacy of a DP mechanism, we compute the *skewness* of the distribution of the estimated $p(x|M(x))$ values. If most words

word	no-noise	Mahalanobis	NADP
misogynist	sexist, chauvinst, bigot	insulting, sexist, racist	scholastic, filicia, shannara
police	officers, cops, authorities	police, authorities, officials	posing, smiling, lying
hitler	adlof, nazi, stalin	hitler, adolf, nazi	paid, ipo, raided
wikileaks	assange, cia, leaked	wikileaks, iran, assange	impetuous, jashari, enraged
FBI	cia, investigation, informant	fbi, history, government	asylum, cardoza, sandiego

Table 1: Top 3 neighbours for words without noise addition to the embeddings (**no-noise**), with SoTA **Mahalanobis** mechanism and the proposed **NADP** mechanism. Mahalanobis mechanism sometimes discloses the original word, whereas NADP mechanism never does.

x_i has lower $p_i = p(x_i|M(x_i))$ values, the probability mass of the p_i distribution will be shifted to the left of the mean, resulting in smaller skewness values (see Appendix B for further explanations). Therefore, smaller skewness values indicate that most words are protected (the probability of being discovered is smaller than the mean) under a DP mechanism.

Results: Figure 3 shows the skewness values reported by Jaccard, Mahalanobis, Gaussian, Laplacian mechanisms and the proposed NADP (for different τ) mechanism for different ϵ values. Overall, we see that NADP reports the lowest skewness values among all DP mechanism compared, indicating that it protects privacy of word embeddings well. We see that the skewness values slightly increase with τ. Recall that when τ increases the similarity of the neighbours connected to a target word by the symmetric neighbouring relation, $\simeq$, increases in the nearest neighbour graph $\mathcal{G}$. Therefore, when τ is high, unless when we apply stronger random noise to word embeddings, it becomes easier to discover the original word from its noise-added embedding. However, we note that the performance of NADP is relatively unaffected by different τ values and skewness values are low for $\tau = 0.1$ setting, which we use in the utility experiments described in § 5.1. Although Gaussian, Jaccard and Mahalanobis mechanisms obtain comparable levels of skewness values when $\epsilon > 15$, for $\epsilon < 5$, where stronger privacy guarantee is required, NADP is the only DP mechanism with near-zero skewness values.

6 Investigating the Nearest Neighbours

To obtain qualitative insights into the levels of privacy provided by NADP, for a given word, we compare its top-3 neighbours in the original embedding space (no-noise added), when Mahalanobis and NADP mechanisms are used to add random noise. Table 1 shows the results for some randomly selected set of words. We see that for the words such as *police*, *hitler*, *wikileaks* and *fbi*, even after applying the Mahalanobis mechanism ($\lambda = 1$), we still retrieve the original word as a nearest neighbour. This indicates that Mahalanobis mechanism is unable to anonymise the target words in these cases. Although not reported here due to space limitations, this problem persists even in Jaccard, Gaussian and Laplace mechanisms, which were under performing to the Mahalanobis mechanism in utility and privacy experiments. In the case of *misogynist*, Mahalanobis mechanism retrieves highly similar neighbours such as *sexist*. On the other hand, the neighbours retrieved from the word embeddings anonymised using NADP are semantically less similar to the target word, thus could be considered to be better preserving the privacy of the target word.

7 Conclusion

We proposed NADP to make word embeddings indistinguishable from their nearest neighbours with theoretical DP guarantees. We compared NADP against existing DP mechanisms in multiple downstream utility experiments which showed its superior performance. Moreover, we evaluated the level of privacy protection provided by NADP against other DP mechanisms. We found NADP to provide stronger privacy guarantees over a broad range of ϵ values. In our future work, we plan to extend NADP to sentence/document embeddings and evaluate for languages other than English.

8 Ethical Considerations

We do not annotate or release any datasets as part of this research. However, the GloVe word embeddings that we use in our experiments are known to contain various types of unfair social biases such as gender and racial biases (Zhao et al., 2018;

Kaneko and Bollegala, 2019, 2021; Gonen and Goldberg, 2019). It is possible that these biases could get further amplified during the neighbourhood computation and noise-addition processes we perform in this work. Therefore, such social biases must be properly evaluated before the noise-added word embeddings produced by our proposed method are used in real-world NLP applications that are used by users.

9 Limitations

Our investigations in this paper was limited to GloVe embeddings, which is one the many avaulable pre-trained static word embeddings. There are other alternative word embeddings such as Skip-Gram with Negative Sampling (SGNS) (Mikolov et al., 2013), PMI-based word embeddings (Arora et al., 2016), fastText embeddings (Bojanowski et al., 2017) etc. that could be used in place of GloVe. However, contextualised word embeddings, obtained using pre-trained Masked Language Models (MLMs) such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), etc. have reported superior performance in various downstream tasks, surpassing that by static word embeddings. Therefore, we consider it to be a natural next step to extend our proposed method to anonymise contextualised word embeddings. The theoretical tools that we develop in this paper should be helpful in proving DP conditions for contextualised word embeddings as well.

All the downstream datasets and word embeddings we considered in this work are limited to the English language, which is known to be a morphologically limited language. Therefore, it is important to evaluate our proposed method on other languages using multilingual word embeddings to verify its effectiveness for the languages other than English.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *CCS*. Association for Computing Machinery, New York, NY, USA, CCS '16, pages 308–318.

Mohamed Abdalla, Moustafa Abdalla, Graeme Hirst, and Frank Rudzicz. 2020. Exploring the privacy-preserving properties of word embeddings: Algorithmic validation study. *JMIR* 22(7):e18055.

Walaa Alnasser, Ghazaleh Beigi, and Huan Liu. 2021. Privacy preserving text representation learning using bert. In *Social, Cultural, and Behavioral Modeling: 14th International Conference, SBP-BRiMS 2021, Virtual Event, July 6–9, 2021, Proceedings*. Springer-Verlag, Berlin, Heidelberg, pages 91–100.

Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *CCS*. Association for Computing Machinery, New York, NY, USA, CCS '13, pages 901–914.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *TACL* 4:385–399.

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

Borja Balle and Yu-Xiang Wang. 2018. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *ICML*. PMLR, volume 80 of *Proceedings of Machine Learning Research*, pages 394–403.

Ghazaleh Beigi, Kai Shu, Ruocheng Guo, Suhang Wang, and Huan Liu. 2019. Privacy preserving text representation learning. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*. Association for Computing Machinery, New York, NY, USA, HT '19, pages 275–276.

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 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)*.

Danushka Bollegala, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2018. Using k-way Co-occurrences for Learning Word Embeddings. In *Proc. of AAAI*. pages 5037–5044.

E. Bruni, N. K. Tran, and M. Baroni. 2014. Multimodal distributional semantics. *JAIR* 49:1–47.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval*. Association for Computational Linguistics, Vancouver, Canada, pages 1–14.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL-HLT*.

Cynthia Dwork and Aaron Roth. 2014. *The Algorithmic Foundations of Differential Privacy*, volume 9. Foundations and Trends in TCS.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 1606–1615.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *WSDM*. Association for Computing Machinery, New York, NY, USA, WSDM '20, pages 178–186.

Oluwaseyi Feyisetan and Shiva Kasiviswanathan. 2021. Private release of text embedding vectors. In *TrustNLP*. Association for Computational Linguistics, Online, pages 15–27.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *EMNLP*. Association for Computational Linguistics, Austin, Texas, pages 2173–2182.

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)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 609–614.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*. PMLR, volume 119 of *Proceedings of Machine Learning Research*, pages 3887–3896.

Prakhar Gupta and Martin Jaggi. 2021. Obtaining better static word embeddings using contextual embedding models. 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)*. Association for Computational Linguistics, Online, pages 5241–5253.

Ivan Habernal. 2021. When differential privacy meets NLP: The devil is in the detail. In *EMNLP*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pages 1522–1528.

Ivan Habernal. 2022. How reparametrization trick broke differentially-private text representation learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Dublin, Ireland, pages 771–777.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD 2004*. pages 168–177.

D. N. Joanes and C. A. Gill. 1998. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)* 47(1):183–189.

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*. Association for Computational Linguistics, Florence, Italy, pages 1641–1650.

Masahiro Kaneko and Danushka Bollegala. 2021. Dictionary-based debiasing of pre-trained word embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, pages 212–223.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2008. What can we learn privately? In *FOCS*. IEEE.

Satyapriya Krishna, Rahul Gupta, and Christophe Dupuy. 2021. ADePT: Auto-encoder based differentially private text transformation. In *EACL*. Association for Computational Linguistics, Online, pages 2435–2439.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *Proc. of ICLR*.

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)*. Association for Computational Linguistics, Melbourne, Australia, pages 25–30.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Lingjuan Lyu, Xuanli He, and Yitong Li. 2020a. Differentially private representation for NLP: Formal guarantee and an empirical study on privacy and fairness. In *EMNLP*. Association for Computational Linguistics, Online, pages 2355–2365.

Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. 2020b. Towards differentially private text representations. In *SIGIR*. Association for Computing Machinery, New York, NY, USA, pages 1813–1816.

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

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proc. of ACL*. pages 115–124.

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

Ryan M Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. 2016. Privacy odometers and filters: Pay-as-you-go composition. In *NeurIPS*. Curran Associates, Inc., volume 29.

Congzheng Song and Vitaly Shmatikov. 2019. Auditing data provenance in text-generation models. In *KDD*. Association for Computing Machinery, New York, NY, USA, KDD '19, pages 196–206.

Gabriel Stanovsky and Mark Hopkins. 2018. Spot the odd man out: Exploring the associative power of lexical resources. In *EMNLP*. Association for Computational Linguistics, Brussels, Belgium, pages 1533–1542.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2049–2054.

Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17(4):395 – 416.

Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong. 2021. Certified robustness to word substitution attack with differential privacy. In *NAACL*. Association for Computational Linguistics, Online, pages 1102–1112.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. In *LREC*.

Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A differentially private text perturbation method using regularized mahalanobis metric. In *PrivateNLP*. Association for Computational Linguistics, Online, pages 7–17.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 4847–4853.

Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. 2020. Private-knn: Practical differential privacy for computer vision. In *CVPR*. pages 11851–11859.

Supplementary Materials

A Downstream Tasks, Datasets and Evaluation Metrics

Word Similarity: The cosine similarity between two words, computed using their word embeddings, is compared against the human similarity ratings using the Spearman correlation coefficient. High degree of correlation with human similarity ratings implies that the word embeddings accurately encode the word-level semantics. We aggregate all of the word-pairs and their human similarity ratings in MEN (Bruni et al., 2014), SimLex (Hill et al., 2015) and SimVerb (Gerz et al., 2016) benchmark datasets and report the overall Spearman correlation in Figure 2a.

Semantic Textual Similarity (STS): In STS, we are provided with sentence-pairs and the human similarity ratings between the two sentences in each pair. Using the word embeddings, we first create an embedding for each sentence and then compute the cosine similarity between the sentence embeddings. The correlation between the predicted sentence similarities and the human ratings is used as the evaluation metric. We represent each sentence by the centroid of the word embeddings corresponding to the words included in that sentence. Although this is a simple method for creating sentence embeddings from word embeddings, it is known to be a strong unsupervised baseline (Arora et al., 2017), and enables us to directly attribute any differences in performance to the word embeddings – the focus in this work. We use the STS Benchmark dataset (Cer et al., 2017), which con-

tains 1379 test sentence-pairs and show the official score (i.e. class-weighted geometric mean of Spearman and Pearson correlation) in Figure 2b.

Text Classification: We train a binary classifier to predict the sentiment (positive vs. negative) of a short review text. Similar to the STS task, we represent a review using the centroid of the word embeddings of the words included in that review. We train a binary logistic regression model to predict sentiment in a review and in Figure 2c report the averaged classification accuracy on the balanced test sets in three standard datasets: Movie reviews dataset (Pang and Lee, 2005), customer reviews dataset (Hu and Liu, 2004) and opinion polarity dataset (Wiebe et al., 2005).

Odd-man-out: Stanovsky and Hopkins (Stanovsky and Hopkins, 2018) proposed the *odd-man-out* task, where given a set of five or more words, a system is required to choose the one which does not belong with the others. They annotated a dataset containing 843 sets via crowd sourcing. Pretrained word embeddings can be used to identify the odd-man in a set by repeatedly excluding one word at a time and measuring the average cosine similarity between all remaining pairs of words. Finally, the word when excluded resulting in the highest pairwise similarity is chosen as the odd-man. Unlike previously described tasks, odd-man-out can be carried out in an unsupervised manner, at word-level, and has higher human agreement between the annotators because it does not require numerical ratings. The percentage of correctly solved sets is shown in Figure 2d.

B Skewness and Privacy

Skewness is a measure of the asymmetry of $p(x|M(x))$ about its mean and can be positive, negative or zero depending on whether $p(x|M(x))$ has respectively a longer left tail, right tail, or perfectly symmetric around the mean (e.g. as in the case of the standard Normal distribution) (Joanes and Gill, 1998). Specifically, if we denote the probability of predicting i-th word x_i by $p_i = p(x_i|M(x_i))$, the skewness of the distribution of p_i over n words is given by $\frac{n}{(n-1)(n-2)} \sum_{i=1}^{n} \left(\frac{p_i - \bar{p}_i}{s}\right)^3$, where $\bar{p}_i$ and s are respectively the mean and standard deviation of $\{p_i\}_{i=1}^{n}$.

We study the relationship between the level of privacy protected by the noise added using a particular DP mechanism, M and the skewness of the distribution of $p(x|M(x))$ for the words w in a vocabulary $\mathcal{X}$. For this purpose, we use the Gaussian mechanism described in § 3.1 in the paper where we sample noise vectors $\boldsymbol{z} \in \mathbb{R}^d$ from the d-dimensional spherical Gaussian $\mathcal{N}(0, \sigma \mathbf{I}_d)$ with zero-mean and standard deviation σ, and add this noise to the word embedding, $\boldsymbol{x} \in \mathbb{R}^d$, representing the word x. Specifically, $M(x) = \boldsymbol{x} + \boldsymbol{z}$. Next, we gradually increase $\sigma \in [0, 1]$ in step size of 0.05 and compute the histograms of $p(x|M(x))$ values for the words in $\mathcal{X}$. The histograms and their skewness values are shown in Figure 4.

From Figure 4, we see that when no-noise is being added (i.e. $\sigma = 0$), the histogram peaks at 1, indicating that all words can be trivially discovered from their word embeddings because the closest neighbour of any target word in the embedding space will be itself. Because the distribution is symmetric around this peak, we have a zero skewness. Overall, we see that when we add increasingly high noise, the histograms start shifting towards to the left because less words will be perfectly discovered from the noise added embeddings. Moreover, we see that more probability mass is distributed towards the right side of the mode (peak), resulting in a longer right tail. Consequently, we see skewness values also continuously increase (except at $\sigma = 0.05$, where the distribution has split into two parts) with σ. This trend stems from the definition of skewness and is independent of the DP mechanism used to generate noise. This result shows that when there are many words with smaller $p(x|M(x))$ values (i.e. distribution has a longer left tail), the skewness values will be smaller, indicating that the privacy is preserved for many words in $\mathcal{X}$.

C Proofs of Theorems

C.1 Proof of Theorem 2

Proof. For any $\varepsilon \geq 0$ and $\delta \in (0, 1)$, put

$$g(u) = \Phi\left(\frac{1}{2u} - \varepsilon u\right) - e^{\varepsilon} \Phi\left(-\frac{1}{2u} - \varepsilon u\right) \quad (7)$$

and $u^* = \min\{u \in \mathbb{R}_{>0} \mid g(u) \leq \delta\}$.

We will prove the following three statements:

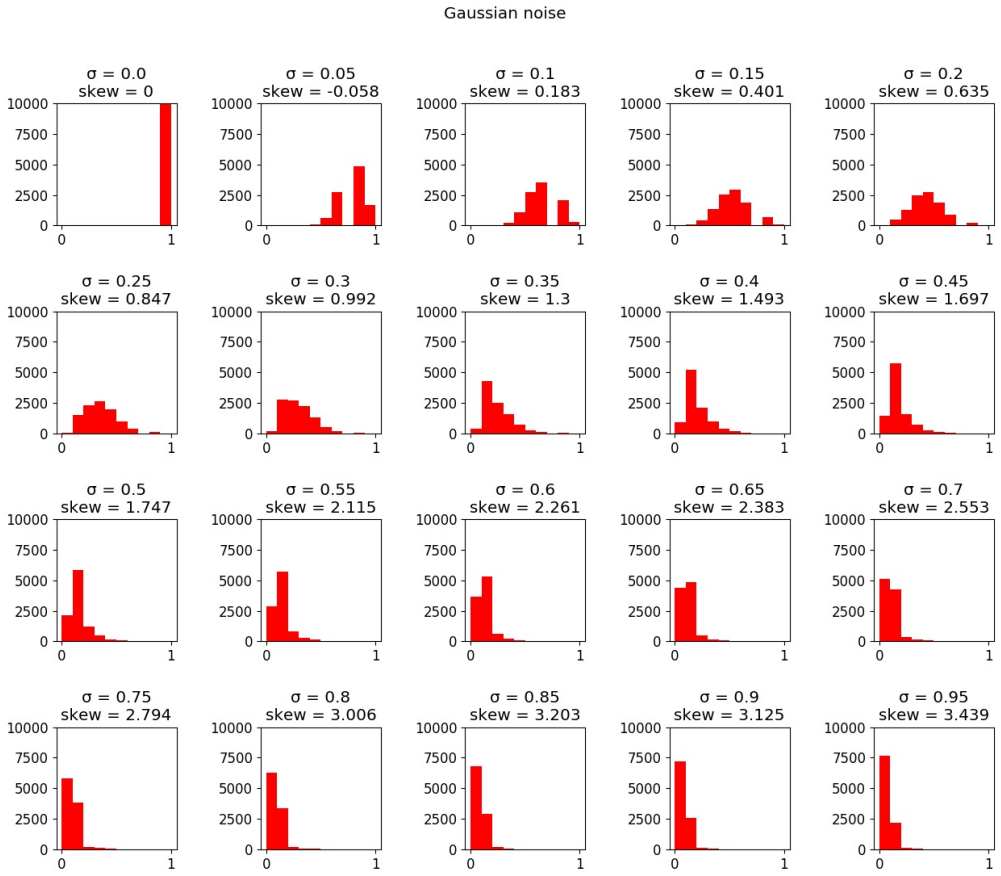


Figure 4: Histogram of p_i values when zero-mean and σ standard deviation Gaussian noise is added to the word embeddings. Skewness values (skew) are shown in each histogram alongside with the σ.

(a) $g(u)$ is strictly decreasing on $(0, \infty)$ with $\lim_{u \to +0} g(u) = 1$, $\lim_{u \to \infty} g(u) = 0$.
(b) The equation $g(u) = \delta$ has the unique solution u^*.
(c) Put $\sigma = u^* \Delta$. Then, the Gaussian output perturbation mechanism $M(x) = \boldsymbol{x} + \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ is (ε, δ)-DP.
To prove (a) Put

$$h(u) = \sqrt{2\pi} g(u) = \sqrt{2\pi} \left\{ \Phi\left(\frac{1}{2u} - \varepsilon u \right) - e^{\varepsilon} \Phi\left(-\frac{1}{2u} - \varepsilon u \right) \right\}.$$

Then,

$$\begin{aligned} h'(u) &= \exp\left(-\frac{1}{2} \left(\frac{1}{2u} - \varepsilon u \right)^2 \right) \cdot \left(-\frac{1}{2u^2} - \varepsilon \right) \\ &\quad - \exp(\varepsilon) \cdot \exp\left(-\frac{1}{2} \left(-\frac{1}{2u} - \varepsilon u \right)^2 \right) \cdot \left(\frac{1}{2u^2} - \varepsilon \right) \\ &= \exp\left(-\frac{1}{2} \left(\frac{1}{2u} - \varepsilon u \right)^2 \right) \cdot \left(-\frac{1}{2u^2} - \varepsilon \right) \\ &\quad - \exp\left(-\frac{1}{2} \left(\frac{1}{2u} - \varepsilon u \right)^2 \right) \cdot \left(\frac{1}{2u^2} - \varepsilon \right) \\ &= -\frac{1}{u^2} \cdot \exp\left(-\frac{1}{2} \left(\frac{1}{2u} - \varepsilon u \right)^2 \right) < 0 \end{aligned}$$

for any $u > 0$. Therefore, $g(u) = (1/\sqrt{2\pi}) h(u)$ is strictly decreasing on $(0, \infty)$. The latter half of the statement is clear from the definition of $g(u)$.
Next, to prove (b) observe that since $g(u)$ is a continuous function on $(0, \infty)$ satisfying $\lim_{u \to +0} g(u) = 1$ and $\lim_{u \to \infty} g(u) = 0$, $g(u) = \delta$ has a solution u' for any $\delta \in (0, 1)$, which must be unique and satisfy $u' = u^*$ because of the monotonicity of $g(u)$.
Finally, to prove (c) let $\sigma = u^* \Delta$. We then have

$$\begin{aligned} &\Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon \sigma}{\Delta} \right) - e^{\varepsilon} \Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon \sigma}{\Delta} \right) \\ &= \Phi\left(\frac{1}{2u^*} - \varepsilon u^* \right) - e^{\varepsilon} \Phi\left(-\frac{1}{2u^*} - \varepsilon u^* \right) = g(u^*) \leq \delta, \end{aligned}$$

which implies the mechanism $M(x) = \boldsymbol{x} + \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ is (ε, δ)-DP as stated in Theorem 2. □

C.2 Proof of Theorem 3

Proof. For any i $(1 \leq i \leq k)$, let $\simeq_i$ be the symmetric neighbouring relation obtained by restricting the relation $\simeq$ on $\mathcal{X}_i$. Then, Δ_i equals to the global L_2 sensitivity of $(\mathcal{X}_i, \simeq_i)$ and $\Delta_{i(x)} = \Delta_i$, $\sigma_{i(x)} = \sigma_i$ for any $x \in \mathcal{X}_i$. Hence, if $\Delta_i > 0$, the mechanism M_i obtained by restricting M on $(\mathcal{X}_i, \simeq_i)$ is (ε, δ)-DP if and only if

$$\Phi\left(\frac{\Delta_{i(x)}}{2\sigma_{i(x)}} - \frac{\varepsilon\sigma_{i(x)}}{\Delta_{i(x)}}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta_{i(x)}}{2\sigma_{i(x)}} - \frac{\varepsilon\sigma_{i(x)}}{\Delta_{i(x)}}\right) \leq \delta$$

for any $x \in \mathcal{X}_i$ by Theorem 2.

Let $x, x' \in \mathcal{X}$ be words such that $x \simeq x'$ and i $(= i(x) = i(x'))$ be the index such that $x, x' \in \mathcal{X}_i$, that is, $x \simeq_i x'$. Now, suppose $\Delta_i > 0$ and (6) holds for any $x \in \mathcal{X}_i$. Then, the mechanism M_i is (ε, δ)-DP and hence, we have

$$\begin{aligned}\mathbb{P}[M(x) \in E] &= \mathbb{P}[M_i(x) \in E] \\ &\leq e^{\varepsilon}\mathbb{P}[M_i(x') \in E] + \delta \\ &= e^{\varepsilon}\mathbb{P}[M(x') \in E] + \delta\end{aligned}$$

for any measurable set $E \subset \mathbb{R}$. Next, suppose $\Delta_{i(x)} = \Delta_{i(x')} = 0$. Then, we have $x = x'$ and hence

$$\mathbb{P}[M(x) \in E] \leq e^{\varepsilon}\mathbb{P}[M(x') \in E] + \delta$$

for any measurable set $E \subset \mathbb{R}$, which implies the mechanism M is (ε, δ)-DP if the condition (6) holds for any $x \in \mathcal{X}$ satisfying $\Delta_{i(x)} > 0$. □

Corollary 1. *For any $\varepsilon \geq 0$ and $\delta \in (0, 1)$, put*

$$g(u) = \Phi\left(\frac{1}{2u} - \varepsilon u\right) - e^{\varepsilon}\Phi\left(-\frac{1}{2u} - \varepsilon u\right)$$

and $u^* = \min\{u \in \mathbb{R}_{>0} \mid g(u) \leq \delta\}$. Also, put $\sigma_{i(x)} = u^*\Delta_{i(x)}$ for any $x \in \mathcal{X}$. Then, the Gaussian output perturbation mechanism $M(x) = \boldsymbol{x} + \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2_{i(x)}\mathbf{I}_d)$ is (ε, δ)-DP.

Proof. Suppose $\Delta_{i(x)} > 0$. Then, by definition, we have

$$\begin{aligned}&\Phi\left(\frac{\Delta_{i(x)}}{2\sigma_{i(x)}} - \frac{\varepsilon\sigma_{i(x)}}{\Delta_{i(x)}}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta_{i(x)}}{2\sigma_{i(x)}} - \frac{\varepsilon\sigma_{i(x)}}{\Delta_{i(x)}}\right) \\ &= \Phi\left(\frac{1}{2u^*} - \varepsilon u^*\right) - e^{\varepsilon}\Phi\left(-\frac{1}{2u^*} - \varepsilon u^*\right) = g(u^*) \leq \delta,\end{aligned}$$

which implies the mechanism $M(x) = \boldsymbol{x} + \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2_{i(x)}\mathbf{I}_d)$ is (ε, δ)-DP as claimed in Theorem 3. □

C.3 Jaccard Mechanism is DP

Theorem 4 claims that the above-mentioned Jaccard mechanism is (ϵ, δ)-DP.

Theorem 4 (Jaccard mechanism is DP)**.** *Jaccard mechanism with $\sigma_i = \Delta\alpha_i\sqrt{2\log(1.25/\delta)}/\epsilon$ is (ϵ, δ)-DP. Here, α_i is a constant that depends only on the density category of a word and Δ is the global sensitivity over the vocabulary.*

Proof. Note that under the Jaccard mechanism, noise vectors, $n(x)$, are sampled from either one of the two Gaussians $\mathcal{N}(\mathbf{0}, \sigma_1\mathbf{I}_d)$ or $\mathcal{N}(\mathbf{0}, \sigma_2\mathbf{I}_d)$ depending on respectively whether $x \in \mathcal{X}_1$ or $x \in \mathcal{X}_2$. Moreover, because α_i depends only on $\mathcal{X}_i$, from $\sigma_i = \Delta\alpha_i\sqrt{2\log(1.25/\delta)}/\epsilon$ and from Theorem 1 we see that each of these underlying Gaussian mechanisms are (ϵ, δ)-DP. Because $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$ by their definitions, it follows from the compositionality property of DP that the overall Jaccard process is also $(0, \delta)$-DP. This proof can be easily extended to more than two density categories by mathematical induction. □

In our experiments, we use $\eta_0 = 6.0$ such that approximately equal numbers of words in $\mathcal{X}$ belong to each category, corresponding to $\alpha_1 = 1.835$ and $\alpha_2 = 1.276$ for $m = 10$. Global sensitivity Δ is computed as the average Euclidean distance between a word and its furthermost neighbour.

The ability to guarantee the mean overlap between neighbourhoods before and after the noise addition is important from the point-of-view of NLP tasks that depend on the neighbourhood information such as semantic similarity measurement, bag-of-words representations-based information retrieval and word/text classification tasks, etc. Unlike in the Gaussian mechanism, in the Jaccard mechanism we have a direct relationship between the level of noise and the performance obtained using the anonymised embeddings in the downstream tasks. Moreover, the Jaccard mechanism allows us to set different noise levels to sparse vs. dense regions in the embedding space, which is not possible with other DP mechanisms.