Private Release of Text Embedding Vectors

Oluwaseyi Feyisetan
Amazon
Seattle, WA, USA
sey@amazon.com

Shiva Kasiviswanathan
Amazon
Palo Alto, CA, USA
kasivisw@amazon.com

Abstract

Ensuring strong theoretical privacy guarantees on text data is a challenging problem which is usually attained at the expense of utility. However, to improve the practicality of privacy preserving text analyses, it is essential to design algorithms that better optimize this tradeoff. To address this challenge, we propose a release mechanism that takes any (text) embedding vector as input and releases a corresponding private vector. The mechanism satisfies an extension of differential privacy to metric spaces. Our idea based on first randomly projecting the vectors to a lower-dimensional space and then adding noise in this projected space generates private vectors that achieve strong theoretical guarantees on its utility. We support our theoretical proofs with empirical experiments on multiple word embedding models and NLP datasets, achieving in some cases more than 10% gains over the existing state-of-the-art privatization techniques.

1 Introduction

Privacy has emerged as a topic of strategic consequence across all computational fields. Differential Privacy (DP) is a mathematical definition of privacy proposed by (Dwork et al., 2006). Ever since its introduction, DP has been widely adopted and as of today, it has become the *de facto* privacy definition in the academic world with also wide adoption in industry, e.g., (Erlingsson et al., 2014; Dajani et al., 2017; Team, 2017; Uber Security, 2017). DP provides provable protection against adversaries with arbitrary side information and computational power, allows clear quantification of privacy losses, and satisfies graceful composition over multiple access to the same data. In DP, two parameters ϵ and δ control the level of privacy. Very roughly, ϵ is an upper bound on the amount of influence a single data point has on the information released and δ is the probability that this bound fails to hold, so the definition becomes more stringent as $\epsilon, \delta \to 0$. The definition with $\delta = 0$ is referred to as *pure differential privacy*, and with $\delta > 0$ is referred to as *approximate differential privacy*.

Within the field of Natural Language Processing (NLP), the traditional approach for privacy was to apply anonymization techniques such as k-anonymity (Sweeney, 2002) and its variants. While this offers an intuitive way of expressing privacy guarantees as a function of an aggregation parameter k, all such methods are provably non-private (Korolova et al., 2009). Given the sheer increase in data gathering occurring across a multiplicity of connected platforms – a great number of which is being done via user generated voice conversations, text queries, or other language based metadata (e.g., user annotations), it is imperative to advance the development of DP techniques in NLP.

Vector embeddings are a popular approach for capturing the "meaning" of text and a form of unsupervised learning useful for downstream tasks. Word embeddings were popularized via embedding schemes such as WORD2VEC (Mikolov et al., 2013), GLOVE (Pennington et al., 2014), and FASTTEXT (Bojanowski et al., 2017). There is also a growing literature on creating embeddings for sentences, documents, and other textual entities, in addition to embeddings in other domains such as in computer vision (Goodfellow et al., 2016).

Recent works such as (Fernandes et al., 2019; Feyisetan et al., 2019, 2020; Xu et al., 2020) have attempted to directly adapt the methods of DP to word embeddings by borrowing ideas from the privacy methods used for map location data (Andrés et al., 2013). In the DP literature, one standard way of achieving privacy is by adding properly calibrated noise to the output of a function (Dwork et al., 2006). This is also the premise behind these previously proposed DP for text techniques, which are based on adding noise to the vector representation of words in a high dimensional embedding space and additional post-processing steps. The

privacy guarantees of applying such a method is quite straightforward. However, the main issue is that the magnitude of the DP privacy noise scales with dimensionality of the vector, which leads to a considerable degradation to the utility when these techniques are applied to vectors produced through popular embedding techniques. In this paper, we seek to overcome this curse of dimensionality arising through the differential privacy requirement. Also unlike previous results which were focused on word embeddings, we focus on the general problem of privately releasing vector embeddings, thus making our scheme more widely applicable.

1.1 Related Work

Vector representations of words, sentences, and documents, have all become basic building blocks in NLP pipelines and algorithms. Hence, it is natural to consider privacy mechanisms that target these representations. The most relevant to this paper is the privacy mechanism proposed in (Feyisetan et al., 2020) that works by computing the vector representation x of a word in the embedding space, applying noise N calibrated to the global metric sensitivity to obtain a perturbed vector $v = x + N$, and then swapping the original word another word whose embedding is closest to v. (Feyisetan et al., 2020) showed that this mechanism satisfies the $(\epsilon, 0)$-Lipschitz privacy definition. However, the issue with this mechanism is that the magnitude (norm) of the added noise is proportional to d, which we avoid by projecting these vectors down before the noise addition step. Our focus here is also more general and not just on word embeddings. Additionally, we provide theoretical guarantees on our privatized vectors. We experimentally compare with this approach.

The privacy mechanisms of (Fernandes et al., 2019; Feyisetan et al., 2019) are also based on similar noise addition ideas. However, (Fernandes et al., 2019) utilized the Earth mover metric to measure distances (instead of Euclidean), and (Feyisetan et al., 2019) perturb vector representations of words in high dimensional Hyperbolic space (instead of a real space). In this paper, we focus on the Euclidean space as it captures the most common choice of metric space with vector models.

Over the past decade, a large body of work has been developed to design basic algorithms and tools for achieving DP, understanding the privacy-utility trade-offs in different data access setups, and on integrating DP with machine learning and statistical inference. We refer the reader to (Dwork and Roth, 2013) for a more comprehensive overview.

Dimensionality reduction for word embeddings using PCA was explored in (Raunak et al., 2019) for computational efficiency purposes. In this paper, we use random projections for dimensionality reduction that helps with reducing the magnitude of noise needed for privacy. Another issue with PCA like scheme is that there are strong lower bounds (that scale with dimension of the vectors d) on the amount of distortion needed for achieving differentially private PCA in the local privacy model (Wang and Xu, 2020).

Random projections have been used as a tool to design differentially private algorithms in other problem settings too (Blocki et al., 2012; Wang et al., 2015; Kenthapadi et al., 2013; Zhou et al., 2009; Kasiviswanathan and Jin, 2016).

2 Preliminaries

We denote $[n] = \{1, \ldots, n\}$. Vectors are in column-wise fashion. We measure the distance between embeddings through the Euclidean metric. For a vector x, we set $\|x\|$ to denote the Euclidean (L_2-) norm and $\|x\|_1$ denotes its L_1-norm. For sets S, T, the Minkowski sum $S + T = \{a + b : a \in S, b \in T\}$. $\mathcal{N}(0, \sigma^2)$ denotes the Gaussian distribution with mean 0 and variance σ^2.

2.1 Privacy Motivations for Text

The privacy concerns around word embedding vectors stem from how they are created. For example, embeddings created using neural models inherit the side effects of unintended memorizations that come with such models (Carlini et al., 2019). Similarly it has been demonstrated that text generation models that encode language representations also suffer from various degrees of information leakage (Song and Shmatikov, 2019; Lyu et al., 2020). While this might not be concerning for off the shelf models trained on public data, it becomes important for word embeddings trained on non-public data.

Recent studies (Song and Raghunathan, 2020; Thomas et al., 2020) have shown that word embeddings are vulnerable to 3 types of attacks (1) *embedding inversion* where the vectors can be used to recreate some of the input training data; (2) *attribution inference* occurs when sensitive attributes (such as authorship) of the input data are revealed even when they are independent of the task at hand;

and (3) *membership inference* where an attacker is able to determine if data from a particular user was used to train the word embedding model.

The privacy consequences are further amplified depending on the domain of data under consideration. For example, a study by (Abdalla et al., 2020) on word embeddings in the medical domain demonstrated that: (1) they were able to reconstruct up to 68.5% of full names based on the embeddings i.e., *embedding inversion*; (2) they were able to retrieve associated sensitive information to specific patients in the corpus i.e., *attribution inference*; and (3) by using the distance between the vector of a patient's name and a billing code, they could differentiate between patients that were billed, and those that weren't i.e., *membership inference*.

These findings all underscore the need to release text embeddings using a rigorous notion of privacy, such as differential privacy, that preserves user privacy and mitigates the attacks described above.

2.2 Background on Differential Privacy.

Differential privacy (Dwork et al., 2006) gives a formal standard of privacy, requiring that, for all pairs of datasets that differ in one element, the distribution of outputs should be similar. In this paper, we use the notion of local differential privacy (LDP) (Kasiviswanathan et al., 2011).

A randomized algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{Z}$ is (ϵ, δ)-local differentially private (LDP) if for any two data $x, x' \in \mathcal{X}$ and all (measurable) sets $U \subseteq \mathcal{Z}$,

$$\mathsf{Pr}[\mathcal{A}(x) \in U] \leq e^{\epsilon}\mathsf{Pr}[\mathcal{A}(x') \in U] + \delta.$$

The probability is taken over the random coins of $\mathcal{A}$. Here, we think of δ as being cryptographically small, whereas ϵ is typically thought of as a moderately small constant. The above definition considers every pair of x and x' (considered as adjacent for the purposes of DP). The LDP notion requires that the given x has a non-negligible probability of being transformed into any other $x' \in \mathcal{X}$ no matter how unrelated (far) x and x' are. However, for text embeddings, this strong requirement makes it virtually impossible to enforce that the semantics of a word are approximately preserved by the privatized vector (Feyisetan et al., 2020). To address this problem, we work with a modification of the above definition, referred to as Lipschitz (or metric) privacy, that is better suited for metric spaces defined through embedding models. Lipschitz privacy is closely related to LDP where the adjacency relation is defined through the Hamming metric, but also generalizes to include Euclidean, Manhattan, and Chebyshev metrics, among others (Chatzikokolakis et al., 2013; Andrés et al., 2013; Chatzikokolakis et al., 2015; Fernandes et al., 2019; Feyisetan et al., 2019, 2020). Similar to differential privacy, Lipschitz privacy is preserved under post-processing and composition of mechanisms (Koufogiannis et al., 2016).

Definition 1 (Lipschitz Privacy (Dwork et al., 2006; Chatzikokolakis et al., 2013))**.** *Let $(\mathcal{X}, d)$ be a metric space. A randomized algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{Z}$ is (ϵ, δ)-Lipschitz private if for any two data $x, x' \in \mathcal{X}$ and all (measurable) sets $U \subseteq \mathcal{Z}$,*

$$\mathsf{Pr}[\mathcal{A}(x) \in U] \leq \exp(\epsilon d(x, x'))\mathsf{Pr}[\mathcal{A}(x') \in U] + \delta.$$

An alternate equivalent way of stating this would be to say that with probability at least $1 - \delta$, over a drawn from either $\mathcal{A}(x)$ or $\mathcal{A}(x')$, we have $|\ln \mathsf{Pr}[\mathcal{A}(x) = a] - \ln \mathsf{Pr}[\mathcal{A}(x') = a]| \leq \epsilon d(x, x')$.

The key difference between Lipschitz privacy and LDP is that the latter corresponds to a particular instance of the former when the distance function is given by $d(x, x') = 1$ for every $x \neq x'$.

In this paper, the metric space of interest is defined by embeddings which organize discrete objects in a continuous real space such that objects that are "similar" result in vectors are "close" in the embedded space. For the distance measure, we focus on the Euclidean metric, $d(x, x') = \|x - x'\|$ that is known to capture semantic similarity between discrete words in a continuous space.

For a function, $f : \mathcal{X} \to \mathbb{R}^m$, the most basic technique in differential privacy to release $f(x)$ is to answer $f(x) + \nu$, where ν is instance-independent additive noise (e.g., Laplace or Gaussian) with standard deviation proportional to the *global sensitivity* of the function f.

Definition 2 (Global sensitivity)**.** *For a function $f : \mathcal{X} \to \mathbb{R}^m$, define the global sensitivity of f as*

$$\Delta_f = \max_{x, x' \in \mathcal{X}} \frac{\|f(x) - f(x')\|}{\|x - x'\|}.$$

2.3 Dimensionality Reduction.

Dimensionality reduction is the problem of embedding a set from high-dimensions into a low-dimensional space, while preserving certain properties of the original high-dimensional set. Perhaps the most fundamental result for dimensionality reduction is the Johnson-Lindenstrauss (JL) lemma which states that any set of p points in high dimensions can be embedded into $O(\log(p)/\beta^2)$ dimensions, while preserving the Euclidean norm of

all points within a multiplicative factor between $1-\beta$ and $1+\beta$. In fact, one could embed an infinite continuum of points into lower dimensions while preserving the Euclidean norm of all point up to a multiplicative distortion. A classical result due to (Gordon, 1988) characterizes the relation between the "size" of the set and the required dimensionality of the embedding on the unit sphere. Before stating the result, we need to introduce the notion of Gaussian width which captures the L_2-geometric complexity of $\mathcal{X}$.

Definition 3 (Gaussian Width)**.** *Given a closed set $\mathcal{X} \subset \mathbb{R}^d$, its Gaussian width $\omega(\mathcal{X})$ is defined as:*

$$\omega(\mathcal{X}) = \mathbb{E}_{g \in \mathcal{N}(0,1)^d} [\sup_{x \in \mathcal{X}} \langle x, g \rangle].$$

Many popular sets have low Gaussian width (Vershynin, 2016). For example, if $\mathcal{X}$ contains vector in $\mathbb{R}^d$ that are c-sparse (at most c non-zero elements) then $\omega(\mathcal{X}) = \sqrt{c \log(d/c)}$. If $\mathcal{X}$ contains vectors that are sparse in the L_1-sense, say $\forall x \in \mathcal{X}$, $\|x\|_1 \leq c$, then $\omega(\mathcal{X}) = O(c\sqrt{\log d})$. Similarly if $\mathcal{X}$ is the d-dimensional probability simplex, then $\omega(\mathcal{X}) = O(\sqrt{\log d})$. Notice that in all these cases $\omega(\mathcal{X})^2$ is exponentially smaller than d.

The following is a restatement of the original Gordon's theorem that is better suited for this paper.

Theorem 1 (Gordon's Theorem (Gordon, 1988))**.** *Let $\beta \in (0,1)$, $\mathcal{X}$ be a subset of the unit d-dimensional sphere and let $\Phi \in \mathbb{R}^{m \times d}$ be a matrix with i.i.d. entries from $\mathcal{N}(0, 1/m)$. Then, $|\|\Phi x\| - 1| \leq \beta$, holds for all $x \in \mathcal{X}$ with probability at least $1 - 2\exp(-\gamma^2/2)$ if $m = \Omega((\omega(\mathcal{X}) + \gamma)^2/\beta^2)$.*

In particular, for a set of points $\mathcal{X} \subset \mathbb{R}^d$, we have the following:
$\mathsf{Pr}\left[\forall x \in \mathcal{X}, |\|\Phi x\| - \|x\|| \leq \beta \|x\|\right] \geq 1 - \gamma$, *if* $m = \Omega((\omega(\mathcal{X}) + \sqrt{\log(1/\gamma)})^2/\beta^2)$

Since for any set $\mathcal{X}$ with $|\mathcal{X}| = p$, $w(\mathcal{X})^2 \leq \log p$, therefore the above theorem is a generalization of the JL lemma. By a simple manipulation and adjusting β, Theorem 1 can be restated for preserving inner-products.

Corollary 2. *Under the setting of Theorem 1, for a set of points $\mathcal{X}$ in $\mathbb{R}^d$,*

$$|\langle \Phi x, \Phi x' \rangle - \langle x, x' \rangle| \leq \beta \|x\| \|x'\|,$$

holds for all $x, x' \in \mathcal{X}$ with probability at least $1-\gamma$, if $m = \Omega((\omega(\mathcal{X}) + \sqrt{\log(1/\gamma)})^2/\beta^2)$.

The above result also holds if we replace the Gaussian random matrix Φ by a sparse random matrix (Bourgain et al., 2015). For simplicity, we use a Gaussian matrix Φ for projection.

3 Our Approach

The main issue arising in constructing differentially private vector embeddings is that a direct noise addition to the vectors (such as in (Feyisetan et al., 2020)) would require that the L_2-norm of the noise vector scales almost linearly with the dimensionality of the vector. To overcome this dimension dependence, our mechanism is based on the idea of performing a dimensionality reduction and then adding noise to the projected vector. By carefully balancing the dimensionality of the vectors with the magnitude of the noise needed for DP, the mechanism achieves a superior performance overall.

We will add noise calibrated to the sensitivity of the dimensionality reduction function. The noise is sampled from a d-dimensional distribution with density $p(z) \propto \exp(-\epsilon\|z\|/\Delta_f)$. Sampling from this distribution is simple as noted in (Wu et al., 2017).[1]. The following simple claim (that holds for all functions f) shows that this mechanism satisfies Definition 1. All the missing proofs from this section are collected in Appendix C.

Claim 3. *Let $f : \mathcal{X} \to \mathbb{R}^m$. Then publishing $\mathcal{A}(x) = f(x) + \kappa$ where κ is sampled from the distribution in $\mathbb{R}^m$ with density $p(z) \propto \exp(-\epsilon\|z\|/\Delta_f)$ satisfies $(\epsilon, 0)$-Lipschitz privacy.*

Let Φ be an $m \times d$ matrix with i.i.d. entries from $\mathcal{N}(0, 1/m)$. Consider an embedding model M. Let $\mathrm{Dom}(M)$ denote the domain and $\mathrm{Ran}(M) \subset \mathbb{R}^d$ denote the range of M. Define a function $f_\Phi : \mathrm{Ran}(M) \to \mathbb{R}^m$ as

$$f_\Phi(x) = \Phi x \text{ and } \Phi \in \mathbb{R}^{m \times d} \text{ i.i.d. from } \mathcal{N}(0, 1/m)\ . \quad (1)$$

Let us first investigate the global sensitivity of f_Φ using Theorem 1. Instead of considering a fixed bound on global sensitivity, we provide a probabilistic upper bound.

Lemma 4. *Let Φ be an $m \times d$ matrix with i.i.d. entries from $\mathcal{N}(0, 1/m)$. Let $\beta \in (0,1)$. If $m = \Omega((\omega(\mathrm{Ran}(M)) + \sqrt{\log(1/\delta)})^2/\beta^2)$, then with probability, at least $1-\delta$, $\Delta_{f_\Phi} \leq 1 + \beta$.*

Let $\beta \in (0,1)$ be a fixed constant. Consider the mechanism which publishes $\mathcal{A}(x) = f_\Phi(x) + \kappa$ where κ is drawn from the distribution with density $p(z) \propto \exp(-\epsilon\|z\|/(1+\beta))$. Given a set of sensitive words $(x_1, \ldots, x_n)$, we can apply $\mathcal{A}(x_i)$ to each word x_i, to release $\mathcal{A}(x_1), \ldots, \mathcal{A}(x_n) \in \mathbb{R}^m$.

[1] The idea is to first sample a uniform vector in the unit sphere in $\mathbb{R}^m$, say v and to sample a magnitude l from the Gamma distribution $\Gamma(m, \Delta_f/\epsilon)$, and output $\kappa = lv$

Algorithm PRIVEMB summarizes the mechanism. Since each vector is perturbed independently, the algorithm can be invoked locally. We now establish the privacy guarantee of PRIVEMB. The δ factor comes in from Lemma 4 because we only have a probabilistic bound on the global sensitivity, i.e., there exists pairs of x, x' for whom the bound on global sensitivity of $1+\beta$ could fail.

For example, imagine a situation where there are n users each having a sensitive word (embedding). Given access to a common Φ, they can perturb their word locally and transmit only the perturbed vector.

Algorithm 1: PRIVEMB

Input: $x_1, \ldots, x_n \in \mathrm{Ran}(M)$ for model M,
privacy parameters $\epsilon, \delta > 0$,
and dimensionality reduction parameter $\beta \in (0,1)$
Output: private vector embeddings $w_1, \ldots, w_n$
Let $m = \Omega((\omega(\mathrm{Ran}(M)) + \sqrt{\log(1/\delta)})^2/\beta^2)$
Let $\Phi \sim_{i.i.d.} \mathcal{N}(0, 1/m)$
for $i \in \{1, \ldots, n\}$ **do**
  $w_i = \Phi x_i + \kappa_i$ where κ_i is i.i.d. from the distr. with density $p(z) \propto \exp(-\epsilon\|z\|/(1+\beta))$
release $(w_1, \ldots, w_n)$

Using Claim 3 and Lemma 4, we now establish that privacy proof for Algorithm PRIVEMB.

Proposition 5. *Algorithm* PRIVEMB *is* (ϵ, δ)*-Lipschitz private. Let* $\beta \in (0,1)$, $\delta > 0$, $\epsilon > 0$, *and* $m = \Omega((\omega(\mathrm{Ran}(M)) + \sqrt{\log(1/\delta)})^2/\beta^2)$. *Let* Φ *be an* $m \times d$ *matrix with i.i.d. entries from* $\mathcal{N}(0, 1/m)$. *Then publishing* $\mathcal{A}(x) = f_\Phi(x) + \kappa$ *where* κ *is drawn from the distribution in* $\mathbb{R}^m$ *with density* $p(z) \propto \exp(-\epsilon\|z\|/(1+\beta))$ *is* (ϵ, δ)*-Lipschitz private.*

It is important to note that the β does not affect the privacy analysis, i.e., for any input parameter β, Algorithm PRIVEMB is (ϵ, δ)-Lipschitz private.

While the idea behind Algorithm PRIVEMB is simple, it is widely applicable and effective. As an example consider vector representation of text such as through Bag-of-K-grams, which creates representations that are sparse in some very high-dimensional space (say c-sparse vectors). In this case, even though d could be extremely large, we can project these vectors to $\approx c\log(d/c)$-dimensional space (due to their low Gaussian width) and add noise in the projected space for achieving privacy. On the other hand, the privacy mechanism of (Feyisetan et al., 2020), with noise magnitude proportional to d will completely destroy the information in these vectors.

4 Utility Analysis of Alg. PRIVEMB

We now provide utility performance bounds for Algorithm PRIVEMB. As mentioned earlier these are the first theoretical analysis for any private vector embedding scheme. We start with two important properties of interest based on distances and inner-products that commonly arise when dealing with text embeddings. Our next result compares the loss of a linear model trained on these private vector embeddings to loss of a similar model trained on the original vector embeddings. All our error bounds depend on $m \approx \omega(\mathrm{Ran}(M))^2$.

We start with a simple observation about the magnitude of the noise vector. Consider κ drawn from the noise distribution with density $p(z) \propto \exp(-\epsilon\|z\|/(1+\beta))$. The Euclidean norm of κ is distributed according to the Gamma distribution $\Gamma(m, (1+\beta)/\epsilon)$ (Wu et al., 2017) and satisfies the following bound.

Claim 6 ((Wu et al., 2017; Chaudhuri et al., 2011))**.** *For the noise vector* κ*, we have that with probability at least* $1-\gamma$, $\|\kappa\| \leq (m\ln(m/\gamma)(1+\beta))/\epsilon$.

Since $\beta < 1$, we can simplify the right hand side of the above claim to $(2m\ln(m/\gamma))/\epsilon$. Let τ be the maximum Euclidean norm of the vectors $x_1, \ldots, x_n$, i.e., $\forall i \in [n], \|x_i\| \leq \tau$.

4.1 Distance Approximation Guarantee

Our first result compares the distances between the private vectors and between the original vectors.

Proposition 7. *Consider Algorithm* PRIVEMB. *With probability at least* $1-\delta$, *for all pairs* $x_i, x_j \in (x_1, \ldots, x_n)$, $|\|w_i - w_j\| - \|x_i - x_j\|| \leq 2\beta\tau + 4(m\ln(2nm/\delta))/\epsilon$.

As a baseline consider the privatization mechanism proposed by (Feyisetan et al., 2020) which computes a privatized version of an embedding vector x by adding noise N to the original vector x. Formally, (Feyisetan et al., 2020) defined a mechanism where the private vector v_i is constructed from x_i as follows: $v_i = x_i + N_i$ where N_i is drawn from the distribution in $\mathbb{R}^d$ with density $p(z) \propto \exp(-\epsilon\|z\|))$ to x. Since the noise vector N_i is now d-dimensional, its Euclidean norm will tightly concentrate around its mean $\mathbb{E}[\|N_i\|] = O(d)$. Therefore, with high probability, $|\|v_i - v_j\| - \|x_i - x_j\|| = \Omega(d)$ holds for the mechanism proposed in (Feyisetan et al., 2020). However, in our mechanism, the dependence on d is replaced by m which as argued above is generally much

smaller than d. On the flip side though, PRIVEMB satisfies (ϵ, δ)-Lipschitz privacy for $\delta > 0$, whereas the mechanism in (Feyisetan et al., 2020) achieves the stronger $(\epsilon, 0)$-Lipschitz privacy.

4.2 Inner-Product Approximation Guarantee

Word embeddings seek to capture word similarity, so similar words (e.g., synonyms) have embeddings with high inner product. We now compare the inner product between the private vectors to the inner product between the original embedding vectors.

Proposition 8. *Consider Algorithm* PRIVEMB. *With probability at least* $1 - \delta$, *for all pairs* $x_i, x_j \in (x_1, \ldots, x_n)$, $|\langle w_i, w_j \rangle - \langle x_i, x_j \rangle| \leq \beta\tau^2 + 8\tau m \ln(2nm/\delta)/\epsilon + (2m\ln(2nm/\delta))^2/\epsilon^2$.

4.3 Performance on Linear Models

We now discuss about the performance of the private vectors $(w_1, \ldots, w_n)$ when used with common machine learning models. Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$ drawn from some universe $\mathbb{R}^d \times \mathbb{R}$ (where y_i represents the label on point x_i), we consider the problem of learning a linear model on this labeled data. We assume that x_i's are sensitive whereas the y_i's are publicly known. Such situations arise commonly in practice. For example, consider a drug company investigating the effectiveness of a drug trail over n users. Here, y_i could represent the response to the drug for user i which is known to the drug company, whereas x_i could encode the medical history of user i which the user would like to keep private.

We focus on a broad class of models, where the loss functions have the form, $\ell(\langle x, \theta \rangle; y)$ for parameter $\theta \in \mathbb{R}^d$, where $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. This captures a variety of learning problems, e.g., the linear regression is captured by setting $\ell(\langle x, \theta \rangle; y) = (y - \langle x, \theta \rangle)^2$, logistic regression is captured by setting $\ell(\langle x, \theta \rangle; y) = \ln(1 + \exp(-y\langle x, \theta \rangle))$, support vector machine is captured by setting $\ell(\langle x, \theta \rangle; y) = \text{hinge}(y\langle x, \theta \rangle)$, where $\text{hinge}(a) = 1 - a$ if $a \leq 1$ and 0 otherwise. We assume that the function ℓ is convex and Lipschitz in the first parameter. Let λ_ℓ denote the Lipschitz parameter of the loss function ℓ over the first parameter, i.e., $|\ell(a; y) - \ell(b; y)| \leq \lambda_\ell |a - b|$ for all $a, b \in \mathbb{R}$.

On the data $(x_1, y_1), \ldots, (x_n, y_n)$, the (empirical) training loss for a parameter θ is defined as: $\frac{1}{n}\sum_{i=1}^n \ell(\langle x_i, \theta \rangle; y_i)$ and the goal in training (empirical risk minimization) is to minimize this loss over a parameter space Θ. Let $\theta^\star$ be a true minimizer of $\frac{1}{n}\sum_{i=1}^n \ell(\langle x_i, \theta \rangle; y_i)$, i.e., $\theta^\star \in \text{argmin}_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^n \ell(\langle x_i, \theta \rangle; y_i)$.

Our goal will be to compare the loss of the model trained on the privatized points $(w_1, y_1), \ldots (w_n, y_n)$ where the w_i's are produced by Algorithm PRIVEMB to the true minimum loss $(= \frac{1}{n}\sum_{i=1}^n \ell(\langle x_i, \theta^\star \rangle; y_i))$. Let $\|\Theta\|$ defined as $\sup_{\theta \in \Theta} \|\theta\|$ denote the diameter of Θ. The following proposition states our result.

Proposition 9. *Consider Algorithm* PRIVEMB. *With probability at least* $1 - \delta$,

$$\min_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^n \ell(\langle w_i, \Phi\theta \rangle; y_i) \leq \frac{1}{n}\sum_{i=1}^n \ell(\langle x_i, \theta^\star \rangle; y_i) + \frac{4\lambda_\ell (m \ln(2nm/\delta))\|\Theta\|}{\epsilon} + \lambda_\ell \beta\tau\|\Theta\|.$$

In the above result the error terms will be negligible if $\beta \ll 1/(\lambda_\ell \tau \|\Theta\|)$ and $\epsilon \gg \lambda_\ell (m \ln(2nm/\delta))\|\Theta\|$. Though in our experiments (see Section 5), we notice good performance with private vectors even when β and ϵ don't satisfy these conditions.

Another point to note is that our setting, where we train ML models over a differentially private data release, is different from traditional literature on differentially private empirical risk minimization where the goal is to release only a private version of model parameter θ, and not the data itself, see e.g., (Chaudhuri et al., 2011; Bassily et al., 2014). In particular, this means that the results from traditional differentially private empirical risk minimization do not carry over to our setting. Our data release setup allows training any number of ML models on the private vectors without having to pay for the cost of composition on the privacy guarantees (as post-processing does not affect the privacy guarantee), which is a desirable property.

5 Experimental Evaluations

We carry out four experiments to demonstrate the improvement of our approach (Algorithm PRIVEMB), denoted as **M2**, over $(\epsilon, 0)$-Lipschitz privacy mechanism proposed in (Feyisetan et al., 2020) (denoted by **M1**).[2] The first three map to the theoretical guarantees described Section 4, i.e., (1) distance approximation guarantee, (2) inner-product approximation guarantee, and (3) performance on linear models. The final experiment provides further evidence for performance of

[2]We choose this mechanism as the baseline as in this setup it achieves the current state-of-the-art utililty guarantees.

using these private vectors for downstream classification tasks. All our experiments are on embeddings generated by GLOVE (Pennington et al., 2014) and FASTTEXT (Bojanowski et al., 2017). The dimensionality of the embedding $d = 300$ in both cases. Due to space constraints, we present the FASTTEXT results in Appendix B.

The value of δ is kept constant for all experiments (involving our scheme) at $1e-6$. We set $\omega(\text{Ran}(M)) = \sqrt{\log d}$. The parameter β only affects the utility guarantee, and Algorithm PRIVEMB is always (ϵ, δ)-Lipschitz private for any value of β. In our experiments, corroborating our theoretical guarantees, we do vary β to illustrate the effect of β on the guarantees. Remember that higher values of β results in lower-dimensional vectors, so setting β appropriately lets one trade-off between the loss of utility due to dimension reduction vs. the gain in the utility due to lesser noise needed for lower-dimensional vectors.

We also vary the privacy parameter ϵ in our experiments. While lower values ϵ are certainly desirable, it is widely known that differentially private algorithms for certain problems (such as those arising in complex domains such as NLP) require slightly larger ϵ values to provide reasonable utility in practice (Fernandes et al., 2019; Feyisetan et al., 2020; Xie et al., 2018; Ma et al., 2020). For example, the related work on differential privately releasing text embeddings from Fernandes *et al.* (Fernandes et al., 2019) and Feyisetan *et al.* (Feyisetan et al., 2020) report values of ϵ of up to 20 and 30 depending on the dimensionality of the space.

5.1 Distance Approximation Guarantees

This experiment compares the distance between pairs of private vectors to that between the corresponding original vectors. We sampled 100 word vectors from the vocabulary. For each of these 100 vectors, we compare the distance to another set of 100 randomly sampled vectors. These 100×100 pair of vectors were kept constant across all experiment runs. For each embedding model, we compared $|\|v_i - v_j\| - \|x_i - x_j\||$ where the v_i's are generated by the schemed in (Feyisetan et al., 2020) (**M1**), to $|\|w_i - w_j\| - \|x_i - x_j\||$ where the w_i's are generated by our scheme (**M2**). The experiments were carried out at values of $\epsilon = 1, 2,$ and 5 for **M1** and **M2**, while varying the values of β for **M2** between 0.5 and 0.7.

Results. The results in Fig. 1 show the experiment outcomes across the different values of ϵ, β, and embeddings. Lower values on the $y-$axis indicate better results in that the distance between the private vectors are a good approximation to the actual distances between the original vectors. Overall, the guarantees of our approach **M2** are better than **M1** as observed by the smaller distance differences across all conditions. Next, the results also highlight that for both mechanisms, as expected, the guarantees get better as ϵ increases, due to the introduction of less noise (note the different scales across ϵ). Finally, the results reveal that for a given value of ϵ, as the value of β increases, the guarantees of our scheme improve. This can be viewed through the guarantees of Proposition 7, which consists of two terms, the first term increases with β and the second term due to its dependence on $1/\beta^2$ (through m) decreases with β. Since the second (noise) term generally dominates, we get an improvement with β, suggesting that it is advantageous to pick a larger β in practice.

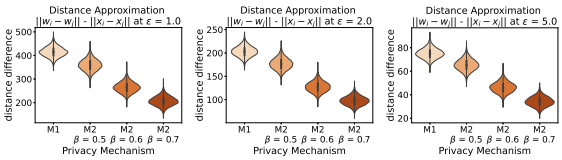

Figure 1: Distance Approximation (GLOVE)

5.2 Inner Prod Approximation Guarantees

This experiment compares the inner product between pairs of private vectors to that between the corresponding original vectors. The setup here is identical to the distance approximation experiments (i.e., the same 100×100 word pairs and mix of ϵ and β). The results capture $|\langle w_i, w_j \rangle - \langle x_i, x_j \rangle|$.

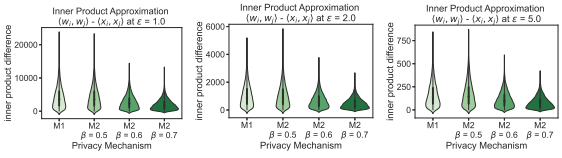

Figure 2: Inner Prod Approximation (GLOVE)

Results. The results in Fig. 2 show the experiment outcomes across ϵ, β, and embeddings. Similar to the findings in Fig. 1, the results of **M2** are an improvement over **M1** with the same patterns of improvement. For a fixed privacy budget, the performance of **M2** is better than that of **M1** and the gap increases as β increases. Again this suggests that one should pick a larger β.

	Non-Private Baselines		**M1**: $\epsilon = 10$		**M2**: $\epsilon = 10, \beta = 0.9$	
Dataset	InferSent	SkipThought	TRAIN ACC	TEST ACC	TRAIN ACC	TEST ACC
MR (Pang and Lee, 2005)	81.10	79.40	58.10	55.61	57.76	58.11
CR (Hu and Liu, 2004)	86.30	83.10	68.32	63.97	72.52	71.02
MPQA (Wiebe et al., 2005)	90.20	89.30	78.76	77.98	77.84	78.86
SST-5 (Socher et al., 2013)	46.30	44.80	31.24	31.90	32.70	32.49
TREC-6 (Li and Roth, 2002)	88.20	88.40	60.54	53.20	62.53	73.00

Table 1: Training and test accuracy scores on classification tasks.

5.3 Performance on Linear Models

We built a simple binary SVM linear mode to classify single keywords into 2 classes: positive and negative based on their conveyed sentiment. The dataset used was a list from (Hu and Liu, 2004) consisting of 4783 negative and 2006 positive words. We selected a subset of words that occurred in both GLOVE and FASTTEXT embeddings and capped both lists to have an equal number of words. The resulting datasets each had 1899 words. The purpose of this experiment was to explore the behaviors of **M1** and **M2** at different values of ϵ and β for a linear model. Results shown are over 10 runs.

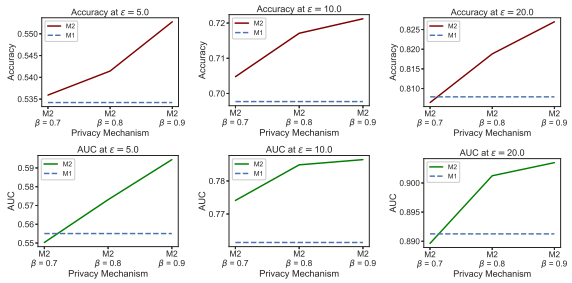

Figure 3: Linear Model Performance (GLOVE)

Results. The results on the performance on linear models are presented in Fig. 3. The performance metrics are (i) accuracy on a randomly selected 20% test set, and (ii) the area under the ROC curve (AUC). Higher values on the $y-$axis indicate better results. The findings follow from our first 2 experiments which demonstrate that for a fixed privacy ϵ guarantee, the utility of **M2** is better than that of **M1** and the gap between the performance of **M2** and **M1** increases as β increases.

5.4 Performance on NLP Datasets

We further evaluated **M2** against **M1** at a fixed value of ϵ and β on classification tasks on 5 NLP datasets. The experiments were done and can be replicated using SentEval (Conneau and Kiela, 2018), an evaluation toolkit for sentence embeddings by replacing the default embeddings with the private embeddings. From the previous experiments, we know that it is better to pick a larger β, so we set $\beta = 0.9$ here.

Results. Table 1 presents the results and summarizes the datasets: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), MPQA (Wiebe et al., 2005), SST-5 (Socher et al., 2013), and TREC-6 (Li and Roth, 2002). Table 1 presents the results from the experiments. We also present results of 2 non-private baselines on all the datasets based on Infersent and SkipThought described in (Conneau et al., 2017). The evaluation metrics were train and test accuracies, therefore, higher scores indicate better utility. Not surprisingly, because of the noise addition there is is a performance drop when we compare the private mechanisms to the non-private baselines. However, the results reinforce our findings that the utility afforded by **M2** are better than **M1** at fixed values of ϵ. Some of the improvements are remarkably significant e.g., $+7\%$ on the CR dataset, and $+20\%$ on TREC-6.

Summary of the Results. Overall, these experiments demonstrate that PRIVEMB offers better utility than the embedding privatization scheme of (Feyisetan et al., 2020).

6 Concluding Remarks

In this paper, we introduced an (ϵ, δ)-Lipschitz private algorithm for generating real valued embedding vectors. Our mechanism works by first reducing the dimensionality of the vectors though a random projection, then adding noise calibrated to the sensitivity of the dimensionality reduction function. The mechanism can be utilized for any well-defined embedding model including but not limited to word, sentence, and document embeddings. We prove theoretical bounds that show how various properties of interest important for vector embeddings are well-approximated through the private vectors, and our empirical results across multiple embedding models and NLP datasets demonstrate the superior utility guarantees.

References

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

Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *ACM CCS*, pages 901–914. ACM.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*. IEEE.

Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2012. The johnson-lindenstrauss transform itself preserves differential privacy. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 410–419. IEEE.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5.

Jean Bourgain, Dirksen Sjoerd, and Jelani Nelson. 2015. Toward a unified theory of sparse dimensionality reduction in euclidean space. In *Proceedings of the 47th ACM Symposium on Theory of Computing*. Association for Computing Machinery.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284.

Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the scope of differential privacy using metrics. In *PETS*.

Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. 2015. Constructing elastic distinguishability metrics for location privacy. *PETS*.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3).

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Aref Dajani, Amy Lauger, Phyllis Singer, Daniel Kifer, Jerome Reiter, Ashwin Machanavajjhala, Simon Garfinkel, Scot Dahl, Matthew Graham, Vishesh Karwa, Hang Kim, Philip Leclerc, Ian Schmutte, William Sexton, Lars Vilhuber, and John Abowd. 2017. The modernization of statistical disclosure limitation at the u.s. census bureau. *Census Scientific Advisory Commitee Meetings*.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer.

Cynthia Dwork and Aaron Roth. 2013. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4).

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *ACM SIGSAC CCS*.

Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. *Principles of Security and Trust*.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *ACM WSDM*.

Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. 2019. Leveraging hierarchical representations for preserving privacy and utility in text. In *IEEE ICDM*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Yehoram Gordon. 1988. *On Milman's inequality and random subspaces which escape through a mesh in* $\mathbb{R}^n$. Springer.

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

Shiva Kasiviswanathan, Homin Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM Journal on Computing*, 40(3).

Shiva Prasad Kasiviswanathan and Hongxia Jin. 2016. Efficient private empirical risk minimization for high-dimensional learning. In *International Conference on Machine Learning*, pages 488–497.

Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. 2013. Privacy via the johnson-lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1):39–71.

Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. 2009. Releasing search queries and clicks privately. In *WebConf*. ACM.

Fragkiskos Koufogiannis, Shuo Han, and George J Pappas. 2016. Gradual release of sensitive data under differential privacy. *Journal of Privacy and Confidentiality*, 7(2).

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING*.

Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. 2020. Towards differentially private text representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1813–1816.

Chuan Ma, Jun Li, Ming Ding, Bo Liu, Kang Wei, Jian Weng, and H. Vincent Poor. 2020. Rdp-gan: A rényi-differential privacy based generative adversarial network.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. *arXiv preprint arXiv:2004.00053*.

Congzheng Song and Vitaly Shmatikov. 2019. Auditing data provenance in text-generation models. In *ACM SIGKDD*.

Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *IJUFKS*, 10(05):557–570.

Apple's Differential Privacy Team. 2017. Learning with privacy at scale. *Apple Machine Learning Journal*, 1(9).

Aleena Thomas, David Ifeoluwa Adelani, Ali Davody, Aditya Mogadala, and Dietrich Klakow. 2020. Investigating the impact of pre-trained word embeddings on memorization in neural networks. In *International Conference on Text, Speech, and Dialogue*, pages 273–281. Springer.

Uber Security. 2017. Uber releases open source project for differential privacy. https://medium.com/uber-security-privacy/7892c82c42b6.

Roman Vershynin. 2016. High dimensional probability. *An Introduction with Applications*.

Di Wang and Jinhui Xu. 2020. Principal component analysis in the local differential privacy model. *Theoretical Computer Science*, 809:296–312.

Yining Wang, Yu-Xiang Wang, and Aarti Singh. 2015. A deterministic analysis of noisy sparse subspace clustering for dimensionality-reduced data. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1422–1431.

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

Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. 2017. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM SIGMOD*, pages 1307–1322.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. 2018. Differentially private generative adversarial network.

Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A differentially private text perturbation method using regularized mahalanobis metric. In *Proceedings of the Second Workshop on Privacy in NLP at EMNLP 2020*, pages 7–17.

Shuheng Zhou, Katrina Ligett, and Larry Wasserman. 2009. Differential privacy with compression. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 2718–2722. IEEE.

Supplementary Material for "Private Release of Text Embedding Vectors"

A Additional Experiments

We now investigate a slightly different setup where we perform the dimensionality reduction while training the embeddings (denoted as **A1**). So here instead of only assuming access to private embeddings vectors as in **M1** and **M2**, we also assume access to the corpus and training platform. Fig. 4 presents results (with linear models as in Experiment 3) on $50d$, $100d$, and $200d$ GLOVE embeddings, and corresponding setting of $\beta = 0.93, 0.66$ and 0.468 in **M2** to match the dimensionality. Unsurprisingly the results below show that **A1** obtains better results than **M2** where the dimensionality reduction happens post training. Mechanism **A1** however has two drawbacks compared to **M2**: (1) it assumes access to the original training corpus and platform which is not always accessible, and (2) it is more computationally expensive as it requires retraining the embeddings from scratch.

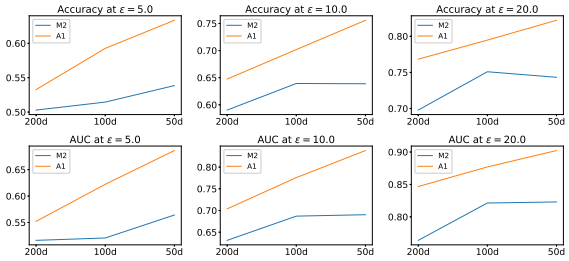

Figure 4: Comparing effects of dimensionality reduction during training vs. after (GLOVE).

B Missing Experiment Results on Fasttext

Experiment 1: Distance Approximation Guarantees.

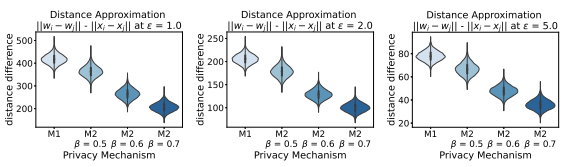

Figure 5: Distance Approximation Experiments (FASTTEXT)

Experiment 2: Inner Prod Approximation Guarantees.

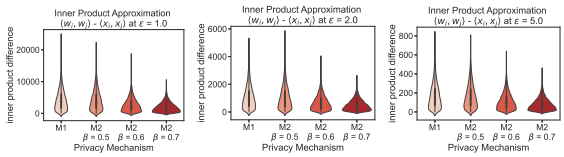

Figure 6: Inner Prod Approximation Experiment (FASTTEXT)

Experiment 3: Performance on Linear Models.

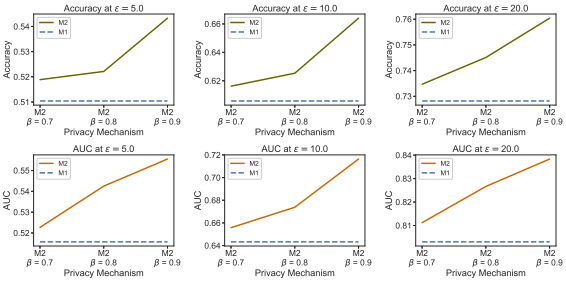

Figure 7: Linear Model Performance Experiments (FASTTEXT)

C Missing Details from Section 3

Claim 10 (Claim 3 Restated)**.** *Let $f : \mathcal{X} \to \mathbb{R}^m$. Then publishing $\mathcal{A}(x) = f(x) + \kappa$ where κ is sampled from the distribution in $\mathbb{R}^m$ with density $p(z) \propto \exp(-\epsilon\|z\|/\Delta_f)$ satisfies $(\epsilon, 0)$-Lipschitz privacy.*

Proof. First note that $f(x) + \kappa$ has the same distribution as that of κ but with a different mean. Consider any $x, x' \in \mathcal{X}$. We will be interested in bounding the ratio $\mathsf{Pr}[\mathcal{A}(x) = w]/\mathsf{Pr}[\mathcal{A}(x') = w]$.

$$\begin{aligned}\frac{\mathsf{Pr}[\mathcal{A}(x) = w]}{\mathsf{Pr}[\mathcal{A}(x') = w]} &= \frac{\exp(-\epsilon\|w - f(x)\|/\Delta_f)}{\exp(-\epsilon\|w - f(x')\|/\Delta_f)} \\ &= \exp(\epsilon(\|w - f(x')\| - \|w - f(x)\|)/\Delta_f) \\ &\leq \exp(\epsilon\|f(x) - f(x')\|/\Delta_f) \\ &\leq \exp(\epsilon\|x - x'\|),\end{aligned}$$

where the first inequality follows from triangle inequality and the last one follows from the definition of global sensitivity (Definition 2). Therefore, for any measurable set $U \subseteq \mathbb{R}^m$, $\mathsf{Pr}[\mathcal{A}(x) \in U] \leq \exp(\epsilon\|x - x'\|)\mathsf{Pr}[\mathcal{A}(x') \in U]$. □

Lemma 11 (Lemma 4 Restated)**.** *Let Φ be an $m \times d$ matrix with i.i.d. entries from $\mathcal{N}(0, 1/m)$. Let $\beta \in (0, 1)$. If $m = \Omega((\omega(\mathrm{Ran}(M)) + \sqrt{\log(1/\delta)})^2/\beta^2)$, then with probability, at least $1 - \delta$, $\Delta_{f_\Phi} \leq 1 + \beta$.*

Proof. Consider the set $\mathrm{Ran}(M) - \mathrm{Ran}(M)$ (where $-$ denotes the Minkowski difference between the sets). By properties of the Gaussian width (see Section 2), the Gaussian width of this new set is at most $\omega(\mathrm{Ran}(M)) + \omega(\mathrm{Ran}(M)) \leq 2\omega(\mathrm{Ran}(M))$. From Theorem 1, under the above setting of m, with probability at least $1-\delta$,

$$\Delta_{f_\Phi} = \max_{x,x' \in \mathrm{Ran}(M)} \frac{\|\Phi x - \Phi x'\|}{\|x - x'\|} \leq (1+\beta).$$

This completes the proof. □

Proposition 12 (Proposition 5 Restated)**.** *Algorithm* PRIVEMB *is (ϵ, δ)-Lipschitz private.*

Proof. Let $\mathcal{A}(x) = f_\Phi(x) + \kappa = \Phi x + \kappa$ where κ is drawn from the distribution in $\mathbb{R}^m$ with density $p(z) \propto \exp(-\epsilon\|z\|/(1+\beta))$. Let $\mathcal{E}$ denote the event that the $\Delta_{f_\Phi} \leq 1-\beta$. From Lemma 11, we know that over the choice of Φ, $\mathsf{Pr}[\mathcal{E}] \geq 1-\delta$. Consider any $x, x' \in \mathrm{Ran}(M)$.

$$\begin{aligned}
&\mathsf{Pr}[\mathcal{A}(x) = w] \\
&= \mathsf{Pr}[\mathcal{A}(x) = w \,|\, \mathcal{E}]\mathsf{Pr}[\mathcal{E}] + \mathsf{Pr}[\mathcal{A}(x) = w \,|\, \bar{\mathcal{E}}]\mathsf{Pr}[\bar{\mathcal{E}}] \\
&\leq \mathsf{Pr}[\mathcal{A}(x) = w \,|\, \mathcal{E}] + \delta,
\end{aligned}$$

where we used that $\mathsf{Pr}[\mathcal{E}] \leq 1$, $\mathsf{Pr}[\mathcal{A}(x) = w \,|\, \bar{\mathcal{E}}] \leq 1$, and $\mathsf{Pr}[\bar{\mathcal{E}}] \leq \delta$. Now under $\mathcal{E}$, from Claim 10, $\mathsf{Pr}[\mathcal{A}(x) = w] \leq \exp(\epsilon\|x - x'\|)\mathsf{Pr}[\mathcal{A}(x') = w]$. Since the above argument holds for all x, x' simultaneously, we get the $\mathcal{A}$ is (ϵ, δ)-Lipschitz private.

Since Algorithm PRIVEMB can be viewed as applying the above mechanism $\mathcal{A}$ on the $x_1, \ldots, x_n$ independently, we get that Algorithm PRIVEMB is (ϵ, δ)-Lipschitz private. □

Proposition 13 (Proposition 7 Restated)**.** *Consider Algorithm* PRIVEMB*. With probability at least $1-\delta$, for all pairs $x_i, x_j \in (x_1, \ldots, x_n)$, $|\|w_i - w_j\| - \|x_i - x_j\|| \leq 2\beta\tau + 4(m\ln(2nm/\delta))/\epsilon$.*

Proof. Let $w_i = \Phi x_i + \kappa_i$ and $w_j = \Phi x_j + \kappa_j$. Using Theorem 1, with probability at least $1-\delta$,

$$\begin{aligned}
&|\|w_i - w_j\| - \|x_i - x_j\|| \\
&= |\|\Phi x_i + \kappa_i - (\Phi x_j + \kappa_j)\| - \|x_i - x_j\|| \\
&\leq |\|\Phi(x_i - x_j)\| - \|x_i - x_j\| + \|\kappa_i\| + \|\kappa_j\|| \\
&\leq \beta\|x_i - x_j\| + \|\kappa_i\| + \|\kappa_j\|. \qquad (2)
\end{aligned}$$

For a fixed i, from Claim 6, we get that with probability at least $1-\delta$, $\|\kappa_i\| \leq (2m\ln(m/\delta))/\epsilon$. Using a union bound,

$$\mathsf{Pr}[\forall i \in [n], \|\kappa_i\| \leq (2m\ln(nm/\delta))/\epsilon] \geq 1-\delta.$$

Plugging this into (2), we get that with probability at least $1-2\delta$, for all $i, j \in [n]$

$$|\|w_i - w_j\| - \|x_i - x_j\|| \leq \beta\|x_i - x_j\| + 4(m\ln(nm/\delta))/\epsilon.$$

Using $\|x_i - x_j\| \leq 2\tau$ and scaling δ completes the proof. □

Proposition 14 (Proposition 8 Restated)**.** *Consider Algorithm* PRIVEMB*. With probability at least $1-\delta$, for all pairs $x_i, x_j \in (x_1, \ldots, x_n)$, $|\langle w_i, w_j\rangle - \langle x_i, x_j\rangle| \leq \beta\tau^2 + 8\tau m\ln(2nm/\delta)/\epsilon + (2m\ln(2nm/\delta))^2/\epsilon^2$.*

Proof. Let $w_i = \Phi x_i + \kappa_i$ and $w_j = \Phi x_j + \kappa_j$. Using Corollary 2, with probability at least $1-\delta$,

$$\begin{aligned}
&|\langle w_i, w_j\rangle - \langle x_i, x_j\rangle| \\
&= |\langle \Phi x_i + \kappa_i, \Phi x_j + \kappa_j\rangle - \langle x_i, x_j\rangle| \\
&= |\langle \Phi x_i, \Phi x_j\rangle + \langle \Phi x_i, \kappa_j\rangle + \langle \kappa_i, \Phi x_j\rangle + \langle \kappa_i, \kappa_j\rangle - \langle x_i, x_j\rangle| \\
&\leq \beta\|x_i\|\|x_j\| + |\langle \Phi x_i, \kappa_j\rangle + \langle \kappa_i, \Phi x_j\rangle + \langle \kappa_i, \kappa_j\rangle| \\
&\leq \beta\|x_i\|\|x_j\| + (1+\beta)\|x_i\|\|\kappa_j\| + (1+\beta)\|x_j\|\|\kappa_i\| + \|\kappa_i\|\|\kappa_j\| \\
&\leq \beta\tau^2 + 2\tau(\|\kappa_j\| + \|\kappa_i\|) + \|\kappa_i\|\|\kappa_j\|. \qquad (3)
\end{aligned}$$

As in Proposition 7,

$$\mathsf{Pr}[\forall i \in [n], \|\kappa_i\| \leq (2m\ln(nm/\delta))/\epsilon] \geq 1-\delta.$$

Plugging this into (3), we get that with probability at least $1-2\delta$, for all $i, j \in [n]$

$$|\langle w_i, w_j\rangle - \langle x_i, x_j\rangle| \leq \beta\tau^2 + \frac{8\tau m\ln(nm/\delta)}{\epsilon} + \frac{(2m\ln(nm/\delta))^2}{\epsilon^2}.$$

By scaling δ we get the claimed bound. □

Proposition 15 (Proposition 9 Restated)**.** *Consider Algorithm* PRIVEMB*. With probability at least $1-\delta$,*

$$\begin{aligned}
\min_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^n \ell(\langle w_i, \Phi\theta\rangle; y_i) &\leq \frac{1}{n}\sum_{i=1}^n \ell(\langle x_i, \theta^\star\rangle; y_i) \\
&+ \frac{4\lambda_\ell(m\ln(2nm/\delta))\|\Theta\|}{\epsilon} + \lambda_\ell\beta\tau\|\Theta\|.
\end{aligned}$$

Proof. By the Lipschitzness assumption,

$$\begin{aligned}
&|\ell(\langle w_i, \Phi\theta^\star\rangle; y_i) - \ell(\langle x_i, \theta^\star\rangle; y_i)| \\
&\leq \lambda_\ell|\langle w_i, \Phi\theta^\star\rangle - \langle x_i, \theta^\star\rangle|. \qquad (4)
\end{aligned}$$

Focusing on the right hand side, from Corollary 2, with probability at least $1-\delta$, for all $i \in [n]$,

$$\begin{aligned}
&|\langle w_i, \Phi\theta^\star\rangle - \langle x_i, \theta^\star\rangle| \\
&= |\langle \Phi x_i + \kappa_i, \Phi\theta^\star\rangle - \langle x_i, \theta^\star\rangle| \\
&\leq |\langle \kappa_i, \Phi\theta^\star\rangle| + \beta\|x_i\|\|\theta^\star\| \\
&\leq (1+\beta)\|\kappa_i\|\|\theta^\star\| + \beta\|x_i\|\|\theta^\star\| \\
&\leq 2\|\kappa_i\|\|\Theta\| + \beta\tau\|\Theta\|,
\end{aligned}$$

where we used $\beta \in (0,1)$, $\|x_i\| \leq \tau$, $\|\theta^\star\| \leq \|\Theta\|$, and with probability at least $1-\delta$, $\|\Phi\theta^\star\| \leq (1+\beta)\|\theta^\star\|$ (from Theorem 1). Using the bound on $\|\kappa_i\|$, we get that with probability at least $1-\delta$, for all $i \in [n]$,

$$|\langle w_i, \Phi\theta^\star\rangle - \langle x_i, \theta^\star\rangle| \leq \frac{4(m\ln(2nm/\delta))\|\Theta\|}{\epsilon} + \beta\tau\|\Theta\|.$$

Plugging this into (4) and averaging over i gives that with probability at least $1-\delta$,

$$\frac{1}{n}\sum_{i=1}^{n} \ell(\langle w_i, \Phi\theta^\star\rangle; y_i)$$
$$\leq \frac{1}{n}\sum_{i=1}^{n} \ell(\langle x_i, \theta^\star\rangle; y_i) + \frac{4\lambda_\ell(m\ln(2nm/\delta))\|\Theta\|}{\epsilon} + \lambda_\ell\beta\tau\|\Theta\|.$$

Since,

$$\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \ell(\langle w_i, \Phi\theta\rangle; y_i) \leq \frac{1}{n}\sum_{i=1}^{n} \ell(\langle w_i, \Phi\theta^\star\rangle; y_i),$$

we get the claimed result. □