TEM: High Utility Metric Differential Privacy on Text

Ricardo Silva Carvalho* Theodore Vasiloudis† Oluwaseyi Feyisetan‡ Ke Wang§

Abstract

Ensuring the privacy of users whose data are used to train Natural Language Processing (NLP) models is necessary to build and maintain customer trust. Differential Privacy (DP) has emerged as the most successful method to protect the privacy of individuals. However, applying DP to the NLP domain comes with unique challenges. The most successful previous methods use a generalization of DP for metric spaces, and apply the privatization by adding noise to inputs in the metric space of word embeddings. However, these methods assume that one specific distance measure is being used, ignore the density of the space around the input, and assume the embeddings used have been trained on public data. In this work we propose Truncated Exponential Mechanism (TEM), a general method that allows the privatization of words using any distance metric, on embeddings that can be trained on sensitive data. Our method makes use of the exponential mechanism to turn the privatization step into a *selection problem*. This allows the noise applied to be calibrated to the density of the embedding space around the input, and makes domain adaptation possible for the embeddings. In our experiments, we demonstrate that our method outperforms the state-of-the-art in terms of utility for the same level of privacy, while providing more flexibility in the metric selection.

1 Introduction

Nowadays, text data are being used as input for a wide variety of machine learning tasks, from next-word prediction in mobile keyboards [10], to critical tasks like predicting patient health conditions from clinical records [22]. Researchers have demonstrated that simple exploratory analysis tasks [14] or the use of models trained on sensitive data [20] may breach the privacy of the individuals involved. Even though there has been research focused on specific privacy-preserving tasks with textual data, such as language models [16, 1, 21], to the best of our knowledge only a more recent line of work [7, 9, 8, 2] has been focusing on providing quantifiable privacy guarantees over the text itself.

In this work, we aim to provide privacy guarantees to textual data using the formal notion of differential privacy (DP) [6], one of the most widely adopted privacy frameworks in industry and academia. More specifically, we apply a generalization called *metric*-differential privacy [3], which allows analysts to tailor solutions over general distance metrics. Previous work [7, 9, 8] in this setting processed each input word by its vector representation and added noise to provide privacy guarantees. Additionally, as a noisy vector is unlikely to exactly represent a valid word, these methods returned a nearest neighbor approximation after querying the representation space. However, these works consider the representation space as non-sensitive, as they do not account for privacy loss in the nearest neighbor search. Moreover, the noise added to a vector does not take into account the density of the region that the vector lies in, which can potentially reduce the utility of a DP algorithm.

Our main contribution is the design of a new mechanism which we call the *Truncated Exponential Mechanism* (TEM), that satisfies metric-DP over textual data, posing the task as a selection problem. Instead of perturbing a representation vector, our method selects an output from a set of possible candidates where words closer to the input word in the metric space have higher probability of being selected. TEM works by adapting the probabilities of words being selected to the regions of a given input word, adjusting the noise injected for better utility, and allows the application of any formal distance function as the metric. The mechanism includes a formal construction with a truncation step to initially select from high utility words with high probability, providing computationally efficient word selection with a tunable error parameter. Our experiments show that TEM obtains higher utility when compared to the state-of-the-art, for the same level of privacy.

2 Related Work

Initially, there has been previous work [7] on text data via the "bag of words" representation of documents,

*Simon Fraser University, Burnaby, Canada. Work done while at Amazon Web Services. Corresponding author: rsilvaca@sfu.ca
†Amazon Web Services, Seattle, United States
‡Amazon, Seattle, United States
§Simon Fraser University, Burnaby, Canada.

applying the Earth Mover's metric to obtain privatized bags, thus performing individual word privatization in the context of metric differential privacy.

Following this context, the Madlib[1] mechanism [8] adds noise to embedding vectors of words, working on in the Euclidean space and adding Laplacian noise to the embedding vectors. After introducing noise, the mechanism outputs the word that is closest to the noisy vector in the embedding space. The algorithm presented in [9] is a follow-up to [8] although it appeared later. This mechanism works in a hierarchical embedding space, where the embedding vector of an input word is perturbed with noise from a hyperbolic distribution. These works successfully illustrated the privacy-utility trade-off on metric differential privacy, and empirically showed that we can achieve reasonable privacy guarantees with the impact on the utility of downstream language models being dependent on the complexity of the downstream task.

We note that the work in [9] compares the hyperbolic mechanism to Madlib [8]. However, since the two algorithms use different metric functions, the evaluation of privacy via only matching the ε parameter of differential privacy can be improved. In this sense, [9] compares the privacy of the two mechanisms, looking at the probability of not changing a word after noise injection, i.e. the probability that the mechanism returns the exact same word used as input. Even though this notion can be intuitively seen as a level of indistinguishability, it cannot guarantee a fair comparison between mechanisms. The issue of comparing metric-DP mechanisms with different metric functions thus remains an open problem. In this work, we only compare mechanisms using the same metric function (Euclidean distance) to ensure a fair comparison.

3 Preliminaries

Consider a user giving as input a word w from a discrete fixed domain $\mathcal{W}$. For any pair of inputs w and w', we assume a distance function $d : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_+$, in a given space of representation of these words. More specifically, we consider a word embedding model $\phi : \mathcal{W} \to \mathbb{R}^n$ will be used to represent words, and the distance function can be a valid metric applicable to the embedding vectors.

Our goal is to select a word from $\mathcal{W}$, based on a given input word, such that the privacy of a user, with respect to this word choice, is preserved. From an attacker's perspective, the output of an algorithm working over input w or w' will become more similar as these inputs become closer with respect to $d(w, w')$. Intuitively, words that are distant in metric space will be more easily distinguishable, compared to words that are close.

With that in mind, we will work on Metric-Differential Privacy [3], a privacy standard defined for randomized algorithms with input from a domain $\mathcal{W}$ that are equipped with a distance metric $d : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_+$ satisfying the three standard axioms of a metric, i.e. formally for all $w, z, y \in \mathcal{W}$:

1. $d(w, z) = 0 \implies w = z$
2. $d(w, z) = d(z, w)$
3. $d(w, y) \leq d(w, z) + d(z, y)$

Thus d as described above using embedding vectors will satisfy all the axioms as long as ϕ is injective.

In this context, the privacy guarantees given by metric-DP depend not only on the privacy parameter ε, but also on the distance metric d used.

Definition 3.1. *(Metric Differential privacy [3]). Given a distance metric $d : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_+$, a randomized mechanism $\mathcal{M} : \mathcal{W} \to \mathcal{Y}$ is εd-differentially private if for any $w, w' \in \mathcal{W}$ and all outputs $y \in \mathcal{Y}$ we have:*

$$\Pr[\mathcal{M}(w) = y] \leq e^{\varepsilon d(w,w')} \Pr[\mathcal{M}(w') = y] \tag{3.1}$$

Usually, on the standard definition of differential privacy[6], the privacy guarantees provided by different mechanisms are compared by looking at the ε value, such that mechanisms with the same ε give the same privacy guarantee. For a fair evaluation on metric-DP using ε, in general, we want to make sure the algorithms we are comparing use the same distance metrics.

For the Euclidean distance metric, as discussed in Section 2, the current state-of-the-art is the Madlib mechanism, which adds Laplacian noise to a given vector in order to obtain a private output. For completion, we detail the Madlib mechanism in Algorithm 1 below.

Algorithm 1 - Madlib: Word Privatization Mechanism for Metric Differential Privacy

Input: Finite domain $\mathcal{W}$, input word $w \in \mathcal{W}$ and privacy parameter ε.
Output: Privatized word.

1: Compute embedding $\phi_w = \phi(w)$
2: Perturb embedding to obtain $\hat{\phi}_w = \phi_w + N$ with noise density $p_N(z) \propto \exp(-\varepsilon \|z\|)$
3: Return perturbed word $\hat{w} = argmin_{y \in \mathcal{W}} ||\phi(y) - \hat{\phi}_w||$

For a Euclidean metric $d : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_+$, Madlib provides metric differential privacy, as formalized next.

[1] We refer to this algorithm with the name used in [4]

THEOREM 3.1. *For a Euclidean metric d, Algorithm 1 is εd-differentially private.*

Following we describe our algorithm that satisfies metric-DP, giving formal proof of its privacy guarantees.

4 Metric Truncated Exponential Mechanism

At its core, our algorithm uses the Exponential Mechanism (EM) [17], which is often used for selection in the context of differential privacy[6].

Algorithm 2, denoted as TEM for metric **T**runcated **E**xponential **M**echanism, is using a variant of the exponential mechanism with Gumbel noise [5], and more specifically adapted to metric-DP for any given distance metric. TEM starts by selecting from words closer to the input by a distance of less or equal to a threshold γ, also including a $\perp$ element to account for the words outside the γ distance.

Algorithm 2 - TEM: Metric Truncated Exponential Mechanism

Input: Finite domain $\mathcal{W}$, input word $w \in \mathcal{W}$, truncation threshold γ, metric $d_{\mathcal{W}} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_+$, and privacy parameter ε.
Output: Privatized word.

1: Given input w, obtain the set L_w such that each word $w_i \in L_w$ satisfies $d_{\mathcal{W}}(w, w_i) \leq \gamma$
2: **for** each $w_i \in L_w$ **do**
3: Set the score $f(w, w_i) = -d_{\mathcal{W}}(w, w_i)$
4: **end for**
5: Create a $\perp$ element with a fixed score defined as: $f(w, \perp) = -\gamma + 2\ln(|\mathcal{W} \setminus L_w|)/\varepsilon$
6: **for** each word $w_i \in L(x) \cup \perp$ **do**
7: Add Gumbel noise with mean 0 and scale $2/\varepsilon$ to the score $f(w, w_i)$
8: **end for**
9: Select $\hat{w}$ as the element with maximum noisy score from $L(x) \cup \perp$
10: **if** $\hat{w} = \perp$ **then**
11: Return random sample of $\mathcal{W} \setminus L_w$
12: **else**
13: Return $\hat{w}$
14: **end if**

To formally state the privacy guarantees of TEM, below we prove that it satisfies εd-differential privacy.

THEOREM 4.1. *For any formal distance metric d, Algorithm 2 is εd-differentially private.*

To prove Theorem 4.1, with the privacy guarantees of TEM, we need to show that the sensitivity of the score function is still $d_{\mathcal{W}}(w, w')$ after the truncation. So first we prove a Lemma giving this result on the sensitivity, and after that, we prove the privacy of TEM.

LEMMA 4.1. *The sensitivity of f is:*

$$\Delta f = \max_{i \in \mathcal{W}} \max_{w, w' \in \mathcal{W}} |f(i, w) - f(i, w')| \leq d_{\mathcal{W}}(w, w')$$

Proof. For a given input w, let us denote $\mathcal{I}(w)$ as the domain elements that have $d_{\mathcal{W}}(i, w) \leq \gamma$, therefore keeping their distances on the score function, while the elements on $\mathcal{W} \setminus \mathcal{I}(w)$ have distances fixed as γ.

In this context, there are four possible cases we need to analyze for a given i and any pair w, w':, which we dive deep into now.

Case 1: $i \in \mathcal{I}(w)$ **and** $i \in \mathcal{I}(w')$. If i is in both $\mathcal{I}(w)$ and $\mathcal{I}(w')$, then it is using its original distance on the score for both w and w'. Thus we have:

$$f(i, w) - f(i, w') = -d_{\mathcal{W}}(i, w) + d_{\mathcal{W}}(i, w') \leq d_{\mathcal{W}}(w, w')$$

with the last inequality being the use of the triangle inequality for the distance metric $d_{\mathcal{W}}$.

Case 2: $i \in \mathcal{I}(w)$ **and** $i \notin \mathcal{I}(w')$. If i is in $\mathcal{I}(w)$ but **not** in $\mathcal{I}(w')$, then it is using its original distance on the score for w and γ for w', therefore:

$$f(i, w) - f(i, w') = -d_{\mathcal{W}}(i, w) + \gamma$$

Since i is **not** in $\mathcal{I}(w')$, it means that $d_{\mathcal{W}}(i, w') > \gamma$, or equivalently $\gamma < d_{\mathcal{W}}(i, w')$, which replacing on the result above gives:

$$f(i, w) - f(i, w') = -d_{\mathcal{W}}(i, w) + d_{\mathcal{W}}(i, w') \leq d_{\mathcal{W}}(w, w')$$

where we use triangle inequality in the last step.

Case 3: $i \notin \mathcal{I}(w)$ **and** $i \in \mathcal{I}(w')$. If i is **not** in $\mathcal{I}(w)$ but is in $\mathcal{I}(w')$, then it is using γ as the distance on score for w and the original distance on score for w', which gives us:

$$f(i, w) - f(i, w') = -\gamma + d_{\mathcal{W}}(i, w')$$

Since i is in $\mathcal{I}(w')$, it means that $d_{\mathcal{W}}(i, w') \leq \gamma$, which replacing on the result above shows:

$$\begin{aligned} f(i, w) - f(i, w') &= -\gamma + d_{\mathcal{W}}(i, w') \\ &\leq -\gamma + \gamma = 0 \leq d_{\mathcal{W}}(w, w') \end{aligned}$$

Case 4: $i \notin \mathcal{I}(w)$ **and** $i \notin \mathcal{I}(w')$. If i is **not** in both $\mathcal{I}(w)$ and $\mathcal{I}(w')$, then it is using γ as distance on score for both w and w', giving:

$$f(i, w) - f(i, w') = -\gamma + \gamma = 0 \leq d_{\mathcal{W}}(w, w')$$

Finally, we note that showing $f(i, w) - f(i, w') \geq -d_{\mathcal{W}}(w, w')$ on the same cases above follows by symmetry.

□

With the sensitivity result, we can now show the privacy guarantee of our mechanism TEM by completing the proof of Theorem 4.1.

Proof. For a given output $y \in \mathcal{W}$ and any pair of inputs $w, w' \in \mathcal{W}$ we have:

$$\frac{\Pr[M(w)=y]}{\Pr[M(w')=y]} = \frac{\exp(\frac{\varepsilon}{2}\cdot f(y,w))}{\exp(\frac{\varepsilon}{2}\cdot f(y,w'))} \cdot \frac{\sum_{z'\in\mathcal{I}(w')}\exp(\frac{\varepsilon}{2}\cdot f(z',w')) + \sum_{w'\in\mathcal{W}\setminus\mathcal{I}(w')}\exp(\frac{\varepsilon}{2}\cdot\gamma)}{\sum_{z\in\mathcal{I}(w)}\exp(\frac{\varepsilon}{2}\cdot f(z,w)) + \sum_{w\in\mathcal{W}\setminus\mathcal{I}(w)}\exp(\frac{\varepsilon}{2}\cdot\gamma)} \tag{4.2}$$

We note that on the second term above we have the same domain of elements, which even though they do not match among summations, they still satisfy the sensitivity on Lemma 4.1, which gives us for the first term on the right-hand side of Equation 4.2:

$$\frac{\exp(\frac{\varepsilon}{2}\cdot f(y,w))}{\exp(\frac{\varepsilon}{2}\cdot f(y,w'))} \leq \frac{\exp(\frac{\varepsilon}{2}\cdot (f(y,w')+d_{\mathcal{W}}(w,w')))}{\exp(\frac{\varepsilon}{2}\cdot f(y,w'))} \leq \exp(\frac{\varepsilon}{2}\cdot d_{\mathcal{W}}(w,w'))$$

And similarly for the second term on the right-hand side of Equation 4.2:

$$\frac{\sum_{z'\in\mathcal{I}(w')}\exp(\frac{\varepsilon}{2}\cdot f(z',w')) + \sum_{w'\in\mathcal{W}\setminus\mathcal{I}(w')}\exp(\frac{\varepsilon}{2}\cdot\gamma)}{\sum_{z\in\mathcal{I}(w)}\exp(\frac{\varepsilon}{2}\cdot f(z,w)) + \sum_{w\in\mathcal{W}\setminus\mathcal{I}(w)}\exp(\frac{\varepsilon}{2}\cdot\gamma)} \leq \exp(\frac{\varepsilon}{2}d_{\mathcal{W}}(w,w'))$$

Therefore, multiplying the two inequalities above gives us for Equation 4.2:

$$\frac{\Pr[M(w)=y]}{\Pr[M(w')=y]} \leq \exp(\frac{\varepsilon}{2}d_{\mathcal{W}}(w,w'))\cdot\exp(\frac{\varepsilon}{2}d_{\mathcal{W}}(w,w')) = \exp(\varepsilon d_{\mathcal{W}}(w,w'))$$

which proves that the mechanism is $\varepsilon d_{\mathcal{W}}$-differentially private.

The only difference in writing from TEM to what we used in the probabilities here is that instead of directly picking from all the elements with distance greater than γ we use $\perp$ first. So now we show they are equivalent.

Let $L(w)$ be a list of elements where each element i satisfies $d_{\mathcal{W}}(i,w) \leq \gamma$ and let $\bar{L}(w)$ be the list of remaining elements $\mathcal{W}\setminus L(w)$. For elements in $\bar{L}(w)$, we see that on TEM they are selected randomly after $\perp$ is selected. Since $\perp$ has score $-\gamma + 2\ln(|\bar{L}(w)|)/\varepsilon$, on the exponential mechanism this is equivalent to:

$$\begin{aligned}\exp(\varepsilon/2\cdot(-\gamma+2\ln(|\bar{L}(w)|)/\varepsilon)) &= \\ \exp(\varepsilon/2\cdot -\gamma)\cdot\exp(\varepsilon/2\cdot(2\ln(|\bar{L}(w)|)/\varepsilon)) &= \\ \exp(\varepsilon/2\cdot-\gamma)\cdot\exp(\ln(|\bar{L}(w)|)) &= \\ \exp(\varepsilon/2\cdot-\gamma)\cdot|\bar{L}(w)|&\end{aligned}$$

This result is essentially the same as using $|\bar{L}(w)|$ elements with score $-\gamma$ directly for selection since, after selecting $\perp$, the elements in $\bar{L}(w)$ are selected randomly, giving each a probability of selection proportional to $\exp(\varepsilon/2\cdot-\gamma)$.

□

4.1 Utility To effectively optimize for utility, next we give a theoretical proposition for defining γ in order to select elements within a γ distance from the input with **high probability**.

THEOREM 4.2. *For $\beta > 0$ and input $w \in \mathcal{W}$, TEM outputs elements with distance less or equal than γ from w with probability at least $1-\beta$ for $\gamma \geq \frac{2}{\varepsilon}\cdot\ln\frac{(1-\beta)(|\mathcal{W}|-1)}{\beta}$.*

Proof. This proof basically uses the fact that on the Exponential Mechanism (EM) [17] we have the probability of a given element proportional to the score of the element.

This theorem is equivalent to guaranteeing that we only output elements outside the γ distance of the input with probability at most β. The worst case to guarantee this condition is when we only have the input word inside the γ distance, and all of the remaining $|\mathcal{W}|-1$ words are outside. Thus we calculate the probability in this theorem for this worst case to obtain the maximum guarantee.

$$\begin{aligned}\frac{(|\mathcal{W}|-1)\cdot\exp(-\varepsilon/2\cdot\gamma)}{\exp(-\varepsilon/2\cdot 0)+(|\mathcal{W}|-1)\cdot\exp(-\varepsilon/2\cdot\gamma)} &\leq \beta \\ \frac{1}{1/\big((|\mathcal{W}|-1)\cdot\exp(-\varepsilon/2\cdot\gamma)\big)+1} &\geq \beta \\ 1/\big((|\mathcal{W}|-1)\cdot\exp(-\varepsilon/2\cdot\gamma)\big)+1 &\leq 1/\beta \\ 1/\big((|\mathcal{W}|-1)\cdot\exp(-\varepsilon/2\cdot\gamma)\big) &\geq (1-\beta)/\beta \\ (|\mathcal{W}|-1)\cdot(1-\beta)/\beta &\leq \exp(\varepsilon/2\cdot\gamma) \\ (2/\varepsilon)\cdot\ln((|\mathcal{W}|-1)\cdot(1-\beta)/\beta) &\leq \gamma\end{aligned}$$

□

The result above gives a guarantee of outputting words that are close to the input w with high probability

for a given γ distance threshold. Thus, it is a theoretical way to choose γ without looking at the data, i.e. without incurring privacy loss, while getting utility guarantees with high probability.

Below we highlight some of the advantages of TEM, specifically compared to the state-of-the-art.

4.2 Detailed comparison with previous work TEM works for any given distance function that satisfies the axioms of a metric. In this sense, it has advantages for future use, when compared to previous mechanisms that used fixed metrics, such as Euclidean [8] and Hyperbolic [9], where changing the metric would need additional complex privacy analyses.

Previous work [7, 8, 9] considered the text privacy preservation problem mainly as a task of releasing a word embedding vector after perturbing with some noise. This means they add noise to each of the dimensions of the vector they aim to release, treating every word embedding vector the same way. In practice, this leads to adding the same amount of noise for any word in the embedding space, regardless of whether the word lies in a dense or sparse region.

In contrast, TEM preserves privacy by posing the task as a *selection problem*, giving words closer to the input word a higher probability of being selected. Therefore, TEM has a more dynamic behavior, adjusting the noise to the density of the domain of selection. In practice, for a given ε, TEM will add less noise to regions with high density, and more noise to regions with low density (sparse), therefore offering better utility in high-density areas.

Moreover, previous work assumed the word embeddings used were trained on a separate **public** dataset, distinct from the data being privatized. TEM does **not** have that requirement, as it can be used safely on sensitive embeddings, providing the potential for further utility gains through the use of domain adaptation, i.e. fine-tuning public pre-trained embeddings on the target **sensitive** data [19, 11].

4.3 Computational Efficiency In terms of computational cost, the bottleneck of previous work resides in the nearest neighbor search of the noisy embedding vector obtained from the input. TEM starts by getting the elements within the distance γ of the input, but instead of querying the nearest neighbor, it queries for neighbors within a given range. In this sense, both methods can rely on fast approximate nearest neighbors implementations that support both querying nearest and by range, such as [12]. Nonetheless, TEM is the only mechanism that, for a fixed domain, is able to pre-process and store the search results for a given γ. After this step, the range search cost becomes constant.

In this context, following we include a rewriting of Algorithm 2 with a pre-processing step that stores results for a given γ. This way, this version of our algorithm has a fast and exact search for the elements within γ distance of any input.

For a fixed finite domain $\mathcal{W}$, given a truncation threshold γ, we pre-compute, for each possible input, the list of elements that satisfy $d_{\mathcal{W}}(i, w) \leq \gamma$, which makes the search for possible candidates of a given input $O(1)$. Another simplification already included in TEM is to use a version of the Exponential Mechanism that uses Gumbel noise [5], which helps avoid dealing with probabilities using the exponential function. Finally, we also point out that we can group all elements below the truncation threshold as one element $\perp$ with the aggregated count, and then if $\perp$ gets selected, randomly sample one of the aggregated elements. Below we give such a simplified algorithm and prove it is equivalent to Algorithm 2.

Algorithm 3 Efficient version of Metric Truncated Exponential Mechanism

Input: Finite domain $\mathcal{W} = \{1, ..., m\}$ of elements, element index $x \in \mathcal{W}$, truncation threshold γ, metric $d_{\mathcal{W}} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$, and privacy parameter ε.
Output: Element index.

Pre-processing:
for each $x \in \mathcal{W}$ **do**
  Create a list $L(w)$ of elements where each element i satisfies $d_{\mathcal{W}}(i, w) \leq \gamma$ and let $\bar{L}(w)$ be the list of remaining elements $\mathcal{W} \setminus L(w)$.
  Define for each $x \in \mathcal{W}$ the score of a $\perp$ element as $f_{\perp}(w) = -\gamma + 2\ln(|\bar{L}(w)|)/\varepsilon$
end for
Selection:
Given input $x \in \mathcal{W}$:
for every element in $L(w) \cup \perp$ **do**
  Add noise from a Gumbel distribution with mean 0 and scale $2/\varepsilon$ to each score $-d_{\mathcal{W}}(\cdot, x)$
end for
Set y as the element with the maximum noisy score
if $y = \perp$ **then**
  Return random sample of $\bar{L}(w)$
else
  Return y
end if

We now formally prove Algorithm 3 is equivalent to Algorithm 2.

LEMMA 4.2. *Algorithm 3 and Algorithm 2 are equal in*

distribution.

Proof. The only difference between the algorithms is that in Algorithm 3 we pre-process $L(w)$ and $\bar{L}(w)$ ahead of time. Since for a fixed domain they do not change and are independent for each input word, the algorithms are equal in distribution. □

Since the two algorithms are equivalent, Algorithm 3 also satisfies the same privacy guarantee.

COROLLARY 4.1. *For any formal distance metric d, Algorithm 3 is εd-differentially private.*

5 Experiments

Now we empirically compare our mechanism TEM given in Algorithm 2 with the current state-of-the-art: the Madlib [8] mechanism.

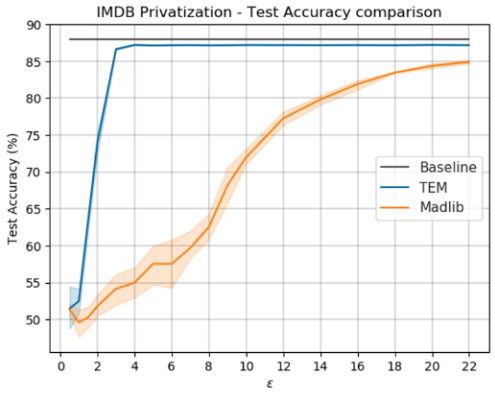
5.1 Settings For fair comparison to Madlib we use TEM with Euclidean distance on a fixed embedding space from GloVe [18]. Experiments use the IMDB reviews dataset [15]. More details are given in Section 5.3.

Utility: To evaluate the utility of the metric-DP mechanisms, we build sentiment classification models on training data privatized by each mechanism and the baseline trained on sensitive data, and compare the accuracy of the trained models on a test dataset.

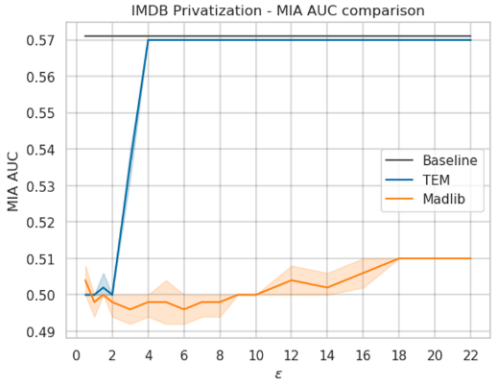
Privacy: As both mechanisms use the Euclidean distance metric, their privacy guarantees are matched by using the same ε. Nonetheless, for illustration, we include the results of a Membership Inference Attack (MIA) [20], which tries to infer the presence of observations used to train a given model based only on black-box access to the model. A lower attack score is better, representing more privacy preservation.

5.2 Results From the results of Figure 1a above we see that for fixed privacy level, TEM outperforms Madlib. More specifically, we see from TEM's results that $\varepsilon \geq 4$ gives a formal level of privacy that is not significant. However, since Madlib adds more noise than needed for different regions of the embedding space, it still achieves some *empirical* privacy, as observed by the MIA results on Figure 1b. Nonetheless, such enforcement is not a formal guarantee of privacy, which for metric-DP is bounded by $\varepsilon d(w, w')$, therefore in this context, it is a loose guarantee of privacy for the embedding space considered.

In this sense, when comparing TEM and Madlib for metric-DP formal levels of privacy, i.e. same ε and metric space, we can clearly see better utility for TEM. Finally, as an example, if we look at $\varepsilon = 2$, where both mechanisms have AUC = 0.50 for MIA,



(a) Utility Evaluation



(b) Empirical Privacy Evaluation

Figure 1: Comparison of mechanisms with 95% confidence interval over 5 trials for various ε. The baseline is built with models trained on original data. TEM used γ from Theorem 4.2 with $\beta = 0.001$.

we see Madlib with average test accuracy of 52% and TEM with 75%, which represents TEM with a relative utility improvement of 42% over Madlib. As discussed previously, by adapting to different densities of the embedding space, TEM is able to add less noise when appropriate, while still giving the same privacy guarantee, which results in the utility gain seen above.

5.3 Reproducibility details Here we describe more settings and include more details about the mechanisms used, to allow reproducibility.

Experiments use the IMDB reviews dataset [15], which gives two different files: training data and testing data, each with 25.000 examples. For the baseline, we trained a model using 50% of the IMDB training data (denote this dataset as TR1) and tested it with 50% of the IMDB testing data (denote this dataset as TE1). For the privatized utility, we trained models on TR1 after

privatization by each mechanism and tested them on the original TE1.

For MIA the models trained as described above were attacked, we denote a given target model as T. For the shadow model, denoted as S, we trained a model on dataset TE2 having the other 50% of the IMDB testing data. To train the attack model, denoted as A, we used as features the output of TE1 and TE2 given by S, where TE2 is labeled as "in" and TE1 as "out". After training model A, we evaluated the inference attack with TR1 and TR2 (having another 50% of the IMDB training data) with features being the output of TR1 and TR2 obtained by a given target model T, where ground-truth for TR1 is "in" and for TR2 is "out".

For embeddings, we used GloVe [18] with 300 dimensions. The sentiment classification models follow the FastText classifier [13], whereas the attack model is an MLP with two layers having 64 hidden nodes each, and ReLU activations. Each model was trained for 20 epochs with a batch size of 64 and default PyTorch parameters for the Adam optimizer.

6 Conclusion

We presented TEM, a mechanism for text privatization on metric differential privacy with formal guarantees. Unlike the current state-of-the-art, our method allows the safe use of sensitive embeddings and provides flexibility in the metric definition. In addition, TEM adapts the noise introduced around regions with different densities to improve utility. Finally, it gives the possibility of performing pre-processing steps for enhanced computational efficiency. Our empirical evaluation demonstrates that TEM obtains better utility than the current state-of-the-art for the same formal privacy guarantees. As future work, we envision the use of domain adaptation, in order to leverage embeddings trained on sensitive data to improve utility. Including the privatization of word context vectors is also a possible enhancement for improved accuracy.

References

[1] Haohan Bo, Steven HH Ding, Benjamin Fung, and Farkhund Iqbal. Er-ae: Differentially-private text generation for authorship anonymization. *arXiv*, pages arXiv–1907, 2019.

[2] Ricardo Silva Carvalho, Theodore Vasiloudis, and Oluwaseyi Feyisetan. Brr: Preserving privacy of text data efficiently on device. *arXiv preprint arXiv:2107.07923*, 2021.

[3] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 82–102. Springer, 2013.

[4] Thomas Diethe. Preserving privacy in analyses of textual data, 2020.

[5] David Durfee and Ryan M Rogers. Practical differentially private top-k selection with pay-what-you-get composition. In *NeurIPS*, pages 3527–3537, 2019.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[7] Natasha Fernandes, Mark Dras, and Annabelle McIver. Author obfuscation using generalised differential privacy. *CoRR*, abs/1805.08866, 2018.

[8] Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 178–186, New York, NY, USA, 2020. Association for Computing Machinery.

[9] Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. Leveraging hierarchical representations for preserving privacy and utility in text. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 210–219. IEEE, 2019.

[10] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[11] Aaron Jaech, Larry Heck, and Mari Ostendorf. Domain adaptation of recurrent neural networks for natural language understanding. *Interspeech 2016*, pages 690–694, 2016.

[12] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[13] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017.

[14] Murat Kantarcioğlu, Jiashun Jin, and Chris Clifton. When do data mining results violate privacy? In *SIGKDD*, pages 599–604. ACM, 2004.

[15] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[16] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models, 2017.

[17] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, volume 7, pages 94–

103, 2007.

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

[19] Barbara Plank and Alessandro Moschitti. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1498–1507, 2013.

[20] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[21] XS Vu, SN Tran, and L Jiang. dpugc: Learn differentially private representation for user generated contents. In *20th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–16, 2019.

[22] Liang Yao, Chengsheng Mao, and Yuan Luo. Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. *BMC medical informatics and decision making*, 19(3):71, 2019.