1-Diffractor: Efficient and Utility-Preserving Text Obfuscation Leveraging Word-Level Metric Differential Privacy

Stephen Meisenbacher
Technical University of Munich
School of Computation, Information and Technology
Garching, Germany
stephen.meisenbacher@tum.de

Maulik Chevli
Technical University of Munich
School of Computation, Information and Technology
Garching, Germany
maulikk.chevli@tum.de

Florian Matthes
Technical University of Munich
School of Computation, Information and Technology
Garching, Germany
matthes@tum.de

ABSTRACT

The study of privacy-preserving Natural Language Processing (NLP) has gained rising attention in recent years. One promising avenue studies the integration of Differential Privacy in NLP, which has brought about innovative methods in a variety of application settings. Of particular note are *word-level Metric Local Differential Privacy (MLDP)* mechanisms, which work to obfuscate potentially sensitive input text by performing word-by-word *perturbations*. Although these methods have shown promising results in empirical tests, there are two major drawbacks: (1) the inevitable loss of utility due to addition of noise, and (2) the computational expensiveness of running these mechanisms on high-dimensional word embeddings. In this work, we aim to address these challenges by proposing 1-Diffractor, a new mechanism that boasts high speedups in comparison to previous mechanisms, while still demonstrating strong utility- and privacy-preserving capabilities. We evaluate 1-Diffractor for utility on several NLP tasks, for theoretical and task-based privacy, and for efficiency in terms of speed and memory. 1-Diffractor shows significant improvements in efficiency, while still maintaining competitive utility and privacy scores across all conducted comparative tests against previous MLDP mechanisms. Our code is made available at: https://github.com/sjmeis/Diffractor.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Natural language processing**;

KEYWORDS

Differential Privacy, Natural Language Processing, Data Privacy

ACM Reference Format:
Stephen Meisenbacher, Maulik Chevli, and Florian Matthes. 2024. 1-Diffractor: Efficient and Utility-Preserving Text Obfuscation Leveraging Word-Level Metric Differential Privacy. In *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics (IWSPA '24), June 21, 2024, Porto, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3643651.3659896

IWSPA '24, June 21, 2024, Porto, Portugal.

ACM ISBN 979-8-4007-0556-4/24/06
https://doi.org/10.1145/3643651.3659896

1 INTRODUCTION

The issue of data privacy has grown in relevance and attention in recent years with respect to the field of Natural Language Processing, particularly with the rising prominence of language models which require significant amounts of data to reach state-of-the-art performance [4, 26]. Tasked with protecting individual privacy while also allowing for the continued proliferation of highly useful models, a number of solutions have appeared in recent literature to form the basis of *privacy-preserving* Natural Language Processing.

One promising and increasingly researched solution comes with the notion of Differential Privacy (DP) [11]. In essence, Differential Privacy provides a mathematically grounded concept of individual privacy protection whose guarantees can be scaled according to the crucial ε parameter, known as the *privacy parameter*. However, the integration of DP into Natural Language Processing does not come without challenges [13, 20, 23], among them the transfer from structured data to textual data and reasoning about the "individual".

In response, one avenue of research looks to the *word-level*, where text is obfuscated via word-by-word replacements, resulting in *perturbed* data [12, 14]. These word-level methods often rely on *Metric Local Differential Privacy* (MLDP), a generalized notion of DP which allows for the extension of DP into metric spaces, such as with words represented in embedding spaces [6]. More recent works have made advancements in the selection strategy of the perturbed word [5], distance metric [15, 38], or calibration of mechanism according to the density of the embedding space [37, 39].

As noted by Mattern et al. [23], a major shortcoming of word-level MLDP methods originates from the relatively large amounts of noise that must be added to satisfy DP, thus ultimately leading to perturbed textual data with poor utility. Another limitation, noted by Klymenko et al. [20], comes with the "structural limitations" imposed by MLDP, particularly when mapping from original text to perturbed text. Such perturbations require nearest neighbor searches, which can become computationally very expensive when working in high-dimensional spaces with large vocabularies.

In this work, we aim to address these two key issues of utility preservation and efficiency with word-level MLDP. To do so, we introduce 1-Dimensional Differentially Private Text Obfuscation (1-Diffractor), a novel method that is highly efficient and boasts competitive levels of utility preservation. In contrast to previous methods, 1-Diffractor operates on single-dimensional word embedding lists and uses the geometric distribution, from which perturbation candidates are selected through what we call a *diffraction* process. An illustration of this process is found in Figure 1 and will be described in Section 3.

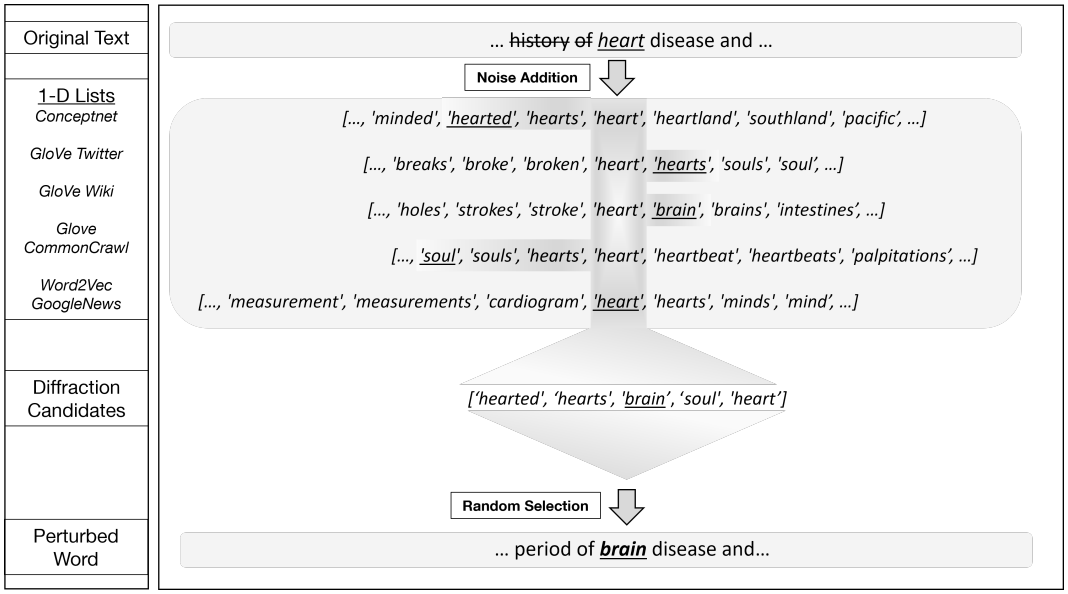


Figure 1: An Overview of 1-Diffractor. Input text is perturbed word-by-word. In this example, we employ the setting in which five word embedding models are used, with one list per model. An input word is *diffracted* through these lists, producing a list of candidate perturbations, from which a final selection is made randomly.

To evaluate 1-Diffractor, we set up three categories of experiments: (1) Utility Experiments, in which two versions of 1-Diffractor are evaluated on the GLUE benchmark, (2) Privacy Experiments, which include a comparative analysis of our mechanism's privacy-preserving capabilities, as well as empirical privacy tests on two adversarial tasks, and (3) Efficiency Experiments, in which the performance and scalability our 1-Diffractor is explored, particularly in comparison to previous mechanisms.

The results of our experiments demonstrate that perturbing datasets with 1-Diffractor preserves utility across a variety of NLP tasks. In addition, 1-Diffractor is effective in reducing adversarial advantage in two chosen tasks. Finally, 1-Diffractor is significantly more efficient than previous methods, processing text at greater than 15x the speed and with less memory than previously.

The contributions of our work are as follows:

(1) We present a novel word-level MLDP mechanism, built upon word embeddings in a one-dimensional space, or *lists*
(2) We demonstrate the effectiveness of our list method with an existing noise-addition mechanism, as well as a new mechanism previously unused in the NLP domain
(3) We also emphasize *efficient* word-level MLDP, highlighting the speed and memory consumption of word perturbations

2 FOUNDATIONS

2.1 Differential Privacy

Intuitively, Differential Privacy (DP) [11] ensures that the result of a computation over a collection is nearly the same irrespective of inclusion or exclusion of a single data point. Hence, if we have two databases $\mathcal{D}$ and $\mathcal{D}'$ differing in only one data point, when a differentially private mechanism is applied over these two databases $\mathcal{D}$ and $\mathcal{D}'$, the result of the mechanism will be very similar. Such databases that differ only by a single element are called *neighboring* or *adjacent* databases. More formally, a mechanism $\mathcal{M} : \mathcal{X}^m \to \mathcal{O}$ operating over any two adjacent databases $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^m$ is (ε, δ)-differentially private, iff $\forall O \subseteq \mathcal{O}$, the following condition holds:

$$\mathbb{P}[\mathcal{M}(\mathcal{D}) \in O] \leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{M}(\mathcal{D}') \in O] + \delta$$

where $\varepsilon > 0$ and $\delta \in [0, 1]$.

In other words, a mechanism is (ε, δ)-differentially private if its output distributions on adjacent databases are "close enough" to each other. In the case of traditional databases, the notion of adjacent databases is simple to understand and without loss of generality could be given as $\mathcal{D}' = \mathcal{D} \cup \{d\}$, where d is a single record in a database. The case of unstructured domains such as text, however, brings additional considerations. Based on how the notion of adjacent databases is defined, so is the element which DP aims to protect. Since this original notion defines databases as those differing in a single record, a differentially private mechanism $\mathcal{M}$ will guarantee that the influence of a single record on the output of the mechanism is bounded.

2.2 Local Differential Privacy and NLP

The notion of DP as introduced above is known as *Global Differential Privacy*. Another notion is called *Local Differential Privacy* (LDP), where noise is added directly to the data before being aggregated at a central location [19]. In LDP, the notion of adjacent databases is defined over data points from a single individual: every collected data point from a single individual is adjacent to every other data point from another individual. Feyisetan et al. [14] leverages the "*one user, one word*" model, where the curator collects a word from each user and uses these to perform some downstream tasks. Instead

of making the original words available to the analyst and leaking information about the users, users can run a privacy-preserving mechanism over their words before releasing them. In the base version of this model, the practicality is quite limited, as collecting a single word from each user would rarely allow for meaningful analysis. As noted by Feyisetan et al. [14], however, this model can be extended to larger textual units (see Section 3.1.3).

Another aspect of this model is the notion of adjacency being defined on words – each word is a database and is adjacent to every other word. However, this is a very strict notion of privacy with severe implications for utility [14]. Hence, a relaxation of (LDP) called Metric Local DP (MLDP) or d_χ-privacy [6] is used instead for a better privacy-utility trade-off [14].

2.3 Metric (Local) Differential Privacy

Let $\mathcal{X}$ and $\mathcal{Z}$ be finite sets and let $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}+$ be the distance metric defined on the set $\mathcal{X}$ that satisfies the axioms of a metric.

Definition 1. *(d_χ-privacy). Let $\varepsilon > 0$. A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Z}$ satisfies εd_χ-privacy iff $\forall x, x' \in \mathcal{X}$ and $\forall z \in \mathcal{Z}$*

$$\frac{\mathbb{P}[\mathcal{M}(x) = z]}{\mathbb{P}[\mathcal{M}(x') = z]} \leq e^{\varepsilon d(x,x')} \tag{1}$$

With MLDP, the notion of adjacent databases remains any two words, but while LDP bounds the output distributions over the adjacent sets by e^ε, MLDP bounds it by $e^{\varepsilon \cdot d_\chi(w,w')}$. Hence, when an MLDP mechanism is applied, the words that are close (measured by some distance metric) would have more "similar" output distributions compared to when the words are far apart. It should be noted that MLDP is a generalized notion of LDP, and (pure) LDP can be derived from MLDP by keeping the distance between any two words as a constant value, i.e., $\forall w, w' \in \mathcal{X}, d_\chi(w, w') = 1$.

Following Feyisetan and Kasiviswanathan [16] and subsequent works, these MLDP mechanisms are run on word *embeddings*, which lend themselves well to the MLDP scheme due to their underlying metrics spaces and distance measures, while still preserving the goal of the *one user, one word* notion.

3 1-DIFFRACTOR

As previously stated, our model operates as defined by Feyisetan et al. [14]: *one user, one word*. In this setting, a utility-preserving private mechanism produces a "perturbed" version of the input word that preserves its original intent but prevents the leakage of the user information that can be extracted from the choice of their word. If the word sent by a user is w, the mechanism $\mathcal{M}$ outputs its "privatized" word $\hat{w} = \mathcal{M}(w)$. Our Mechanism $\mathcal{M}$ operates on word embeddings where the words are arranged in a one-dimensional list, with adjacent words being close in the original space.

Intuition behind converting a word embedding model from $\mathbb{R}^d$ to $\mathbb{Z}$. Previous word-level MLDP mechanisms operate on high-dimensional word embeddings, adding noise to every dimension of the vector. This not only adds a high value of noise (measured by its norm), but the noisy vector rarely corresponds to a word in the embedding model. Hence, it must be remapped to a nearby word, increasing the time complexity and potentially impacting the utility of the overall mechanism. There exist multiple approaches to remapping the vector to a word [14, 37, 39]. Several approaches that do not add noise directly to word vectors have been proposed that use variants of the Exponential Mechanism for choosing a privatized word for the input word [5, 36]. We formulate our mechanism in a different way that is fast to compute, by reducing the dimension of embeddings to one dimension. As such, noise must only be added on one dimension; concretely, we add discretized noise sampled from Geometric distribution to words in one-dimensional space.

Converting high-dimensional embeddings to a 1-D list has several advantages: 1-D embeddings can be considered an index and this simplifies word privatization to returning a *noisy index*. Moreover, these 1-D lists can be combined together into a collection of lists from different embedding models, thereby increasing the diversity of output words while also providing an extra layer of obfuscation, described further in Section 3.1.2.

To create such a one-dimensional list from a word embedding model such as word2vec [25] or GloVe [27], we initialize the list L with a random word and iteratively add the nearest word in the embedding space to the previously added word. Concretely, a random word is first selected as the seed word and it is added to the list L. Then the nearest word in the embedding model, according to the Euclidean distance, is made the seed word and the process repeats for the remaining embedding space, until no words remain.

Thus, this process mimics a "greedy search" through a given embedding space, from a randomly selected starting point. Algorithm 1 outlines this process. Multiple lists from a single embedding model can be created by initializing the starting point (seed) at different points. In addition, various pre-trained word embedding models can be utilized in tandem to create several lists.

Algorithm 1
Creation of a word list L from a word embedding model

```
Require: Word Embedding model E
  L ← list()
  words ← vocabulary(E)
  seed ← random(words)
  L.append(seed)
  while words.length() > 1 do
      N ← NearestWord(seed)
      L.append(N)
      words.remove(seed)
      seed ← N
  end while
  return L
```

3.1 Word-level d_χ-privacy mechanism

3.1.1 Using a single word embedding list. We describe how our proposed d_χ-privacy mechanism works with a single word embedding list L. We define an embedding function $\Phi : \mathcal{V} \to [0, |\mathcal{V}|] \cap \mathbb{N}$ over the list L that takes a word from our vocabulary set $\mathcal{V}$ as input and returns its position (index) in the list as the output. Using these indices, we define a distance function $d_\mathcal{V} : \mathcal{V} \times \mathcal{V} \to \mathbb{Z}+$ that gives us the distance between two words w and w' in our list as follows:

$$d_\mathcal{V}(w, w') = |\Phi(w) - \Phi(w')| \tag{2}$$

Note that the $d_\mathcal{V}$ distance function indeed follows all three axioms of a distance metric. Now we define our mechanism that

takes a word w as input and outputs a "privatized" word w' using $d_{\mathcal{V}}$-privacy mechanism.

The mechanism $\mathcal{M} : \mathcal{V} \rightarrow \mathbb{Z}$ operates over a word and adds noise sampled from the geometric distribution. This particular distribution can be utilized due to our representation of words in one dimension (discrete indices), and thus we return a noisy index. Mathematically, $\mathcal{M}$ is defined as follows:

$$\mathcal{M}(w, \Phi(\cdot), \varepsilon) = \Phi(w) + x, x \sim \mathcal{G}\left(0, \frac{1}{\varepsilon}\right) \tag{3}$$

where $\mathcal{G}$ is the Geometric distribution, given by the following probability density function,

$$\forall x \in \mathbb{Z}, \quad \mathop{\mathbb{P}}_{X \leftarrow \mathcal{G}(\mu, b)}[X = x] = \frac{e^{1/b} - 1}{e^{1/b} + 1} \cdot e^{-\frac{|x-\mu|}{b}} \tag{4}$$

Using the property of linear transformation of random variables, one can see that our mechanism $\mathcal{M}$ is a randomized algorithm and its outputs are random variables drawn from the Geometric distribution $\mathcal{G}(\Phi(w), 1/\varepsilon)$. The proof that $\mathcal{M}$ satisfies $\varepsilon d_{\mathcal{V}}$-privacy can be found in Theorem 1.

Theorem 1. *The proposed mechanism $\mathcal{M}$ defined in Equation 3 satisfies $\varepsilon d_{\mathcal{V}}$-privacy.*

Proof. Let w and w' be any two words belonging to set $\mathcal{V}$, then the ratio of the probability distribution of application of $\mathcal{M}$ on w and w' can be given as

$$\begin{aligned}
\frac{\mathbb{P}[\mathcal{M}(w) = x]}{\mathbb{P}[\mathcal{M}(w') = x]} &= \frac{e^{-\varepsilon \cdot |x - \Phi(w)|}}{e^{-\varepsilon \cdot |x - \Phi(w')|}} \\
&\quad \text{(From Equation 4)} \\
&= e^{\varepsilon \cdot (|x - \Phi(w')| - |x - \Phi(w)|)} \\
&\leq e^{\varepsilon \cdot (|\Phi(w) - \Phi(w')|)} \\
&\quad (|a| - |b| \leq |a - b|) \\
&= e^{\varepsilon \cdot d_V(w, w')} \\
&\quad \text{(From Equation 2)}
\end{aligned}$$

□

Now, we define a truncation function $t : \mathbb{Z} \rightarrow [0, |\mathcal{V}|] \cap \mathbb{N}$ that takes the input from our mechanism defined above and truncates its output to the range $\{0 \ldots |\mathcal{V}|\}$ in the following way:

$$t(x) = \begin{cases} x & x \in [0, |\mathcal{V}|] \\ 0 & x < 0 \\ |\mathcal{V}| & x > |\mathcal{V}| \end{cases} \tag{5}$$

The application of the function $t(\cdot)$ to the output of the mechanism $\mathcal{M}$ truncates its values in the range $[0, |\mathcal{V}|]$. Due to resilience to post-processing of $\varepsilon d_{\mathcal{X}}$-privacy, the composition $(t \circ \mathcal{M})$ also satisfies $\varepsilon d_{\mathcal{V}}$-privacy [21]. Alternatively, the composition $(t \circ \mathcal{M})$ can be thought of as a randomized mechanism $\mathcal{M}'$ that adds to the index $\Phi(w)$ a random variable x drawn from a *Truncated Geometric* distribution instead of the *Geometric* distribution in Equation 3.

In order to convert the privatized index back to the domain of words, we can apply a function $r : [0, |\mathcal{V}|] \cap \mathbb{N} \rightarrow \mathcal{V}$. Note that function r is the inverse function of our embedding function $\Phi(\cdot)$. $r(\cdot)$ takes the index of the word and returns its corresponding word. Again from the post-processing property of metric-DP, the composition $(r \circ t \circ \mathcal{M})$ satisfies $\varepsilon d_{\mathcal{V}}$-privacy [21]. Hence, our function $(r \circ t \circ \mathcal{M}) : \mathcal{V} \rightarrow \mathcal{V}$ takes a word and outputs a "privatized" word.

3.1.2 Using multiple word embedding lists. One can use multiple lists $\mathbb{L} = \{L_1, L_2, \ldots, L_n\}$ as well. We define separate word embedding functions $\Phi^l : \mathcal{V} \rightarrow [0, |\mathcal{V}|] \cap \mathbb{N}$ corresponding to each list L_l and by extension, separate distance functions $d^l_{\mathcal{V}} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{Z}+$ for each list L_l. Then, on the input word, we apply the mechanism $\mathcal{M}(w, \Phi^l(\cdot), \varepsilon)$ for every list in $\mathbb{L}$, which outputs perturbed words $\mathbb{W} = \{w'_1, w'_2, \ldots, w'_n\}$ for an input word, and we **randomly** select a single word out of $\mathbb{W}$, releasing that as the "privatized" word for the input word. Even though the mechanism is applied n times, only a single output out of all n results is released, implying it is not a sequential application of $d_{\mathcal{X}}$-privacy mechanism; hence, we do not incur any additional privacy cost in terms of ε as compared to utilizing a single list. However, since the distance between the two words w and w' may not be the same across lists, the probability distributions resulting from the application of mechanism $\mathcal{M}$ on w and w' would be bounded by $e^{\varepsilon \cdot d_{max}(w, w')}$, i.e.,

$$\frac{\mathbb{P}[\mathcal{M}(w) = x]}{\mathbb{P}[\mathcal{M}(w') = x]} \leq e^{\varepsilon d_{max}(w, w')} \tag{6}$$

where $d_{max}(w, w') = \max_{l \in \{1 \ldots n\}} d^l_{\mathcal{V}}(w, w')$.

3.1.3 Extending Word-level $d_{\mathcal{X}}$-privacy to sentences. Our proposed mechanism 1-Diffractor operates on the word level; however, the perturbation of large units of textual data, i.e., sentences can be extrapolated from this base level, as described in [14].

In particular, a sentence s can be considered as a concatenation of n words, i.e., $s = w_1 \cdot w_2 \cdots w_n$. We follow [14] and apply our mechanism to each word independently to generate a privatized sentence $s' = x_1 \cdot x_2 \cdots x_n$. We define the distance function $D : \mathcal{V}^* \times \mathcal{V}^* \rightarrow \mathbb{Z}+$ between two sentences of same length $s = w_1 \cdots w_n$ and $s' = w'_1 \cdots w'_n$ as $\mathcal{D} = \sum_{i=1}^{n} d_{max}(w_i, w'_i)$..

Since the application of the mechanism $\mathcal{M}$ is independent with respect to words, when $\mathcal{M}$ is applied on a sentence $s = w_1 \cdots w_n$, its output distribution is given by:

$$\mathbb{P}[\mathcal{M}(s) = z] = \prod_{i=1}^{n} \mathbb{P}[\mathcal{M}(w_i) = x]$$

Theorem 2. *$\forall s, s' \in \mathcal{V}^*, \forall z \in \mathcal{V}^*$ and $|s| = |s'| = |z| = n$, we have the following inequality:*

$$\frac{\mathbb{P}[\mathcal{M}(s) = z]}{\mathbb{P}[\mathcal{M}(s') = z]} \leq \exp\left(\varepsilon \cdot \mathcal{D}(s, s')\right)$$

Proof.

$$\begin{aligned}
\frac{\mathbb{P}[\mathcal{M}(s) = z]}{\mathbb{P}[\mathcal{M}(s') = z]} &= \prod_{i=1}^{n} \frac{\mathbb{P}[\mathcal{M}(w_i) = x]}{\mathbb{P}[\mathcal{M}(w'_i) = x]} \\
&\leq \prod_{i=1}^{n} \exp\left(\varepsilon \cdot d_{max}(w_i, w'_i)\right) \\
&= \exp\left(\varepsilon \cdot \sum_{i=1}^{n} d_{max}(w_i, w'_i)\right) \\
&= \exp\left(\varepsilon \cdot \mathcal{D}(s, s')\right)
\end{aligned}$$

□

The limitation of extending word-level metric-DP to a sentence is that the neighboring "dataset" should be sentences of the same

length, and hence the output sentence actually leaks the number of words in a sentence, as pointed out by Mattern et al. [23].

4 UTILITY EXPERIMENTS

To test the utility-preservation of 1-Diffractor, we have designed a three-part experiment, consisting of (1) experiments with different settings of our method on the GLUE benchmark, (2) comparative tests on selected GLUE tasks, comparing our method against previous MLDP mechanisms, and (3) a semantic similarity test to evaluate how well 1-Diffractor preserves meaning.

4.1 Design

We decide to evaluate utility on the GLUE benchmark [34], which presents a series of nine NLP tasks broken down into three categories. These include binary classification tasks (CoLA, SST2), textual similarity tasks (QQP, MRPC, STSB), and textual entailment tasks (MNLI, QNLI, WNLI, RTE). Previous works on text-to-text privatization perform evaluations on single tasks for the benchmark [32, 40], but to the best of the authors' knowledge, no works have done so for the entire benchmark. Of particular note is the difficulty introduced when a benchmark dataset contains two sentences per data point, which presents an interesting test case for the utility-preserving capabilities of an MLDP mechanism.

4.1.1 Dataset Preparation. For each dataset in the GLUE benchmark, we perturb all relevant columns with 1-Diffractor in both the train and validation splits, i.e., either a single sentence or a sentence pair. For the larger datasets in the benchmark (MNLI, QNLI, QQP, SST2), we take 10% of the training dataset. This is justified due to the large size of these datasets, as well as the fact that all experiments report *relative* results to the baseline. In all cases, the full validation set is used. For MNLI, only the *matched* split is used.

4.1.2 Baseline Model and Scoring. For all utility tests, we fine-tune BERT (bert-base-uncased) [9] on the train split, and report the evaluation performance on the validation split. For both the original dataset and all subsequent evaluations (perturbed datasets), the fine-tuning process is run on a V100 GPU (Google Colab) and is repeated **three** times, for one epoch each. This is to account for variations in the training process. Final scores are calculated by averaging the accuracy scores for the three runs, while also calculating the standard deviation. All tasks report accuracy scores, except for STSB, where the Pearson-Spearman Correlation is reported.

4.1.3 Experiment Parameters.

Noise mechanism. We choose: (1) the *Truncated Geometric* mechanism, as introduced in Section 3.1, denoted as 1-D$_G$, and (2) the *Truncated Exponential* mechanism (TEM), introduced by Carvalho et al. [5], denoted as 1-D$_T$. In the case of 1-D$_T$, we adapt the usage of TEM for one-dimensional space, so as to fit 1-Diffractor. In particular, 1-D$_T$ operates on a subset of the vocabulary, governed by a truncation threshold γ, whose value depends on the chosen ε.

Choice of epsilon (ε). We choose the values $\varepsilon \in \{0.1, 0.5, 1, 3, 5, 10\}$, which upon initial observation of our method, represents the "effective range" of noise addition, thus allowing for a test of strict privacy guarantees (e.g., 0.1) as well as weaker guarantees (e.g., 10). As will be noted in Section 5.3, this range is extended for comparative tests.

1-Diffractor settings. We utilize five embedding models[1] and three list configurations:

- E1: *conceptnet-numberbatch* [31]
- E2: *glove-twitter* [27]
- E3: *glove-wiki-gigaword* [27]
- E4: *glove-commoncrawl* [27]
- E5: *word2vec-google-news* [25]
- **L0**: 1 list for each of E1-5
- **L1**: 2 lists for each of E1-5
- **L2**: 1 list, only E1

4.2 Results

Table 1 presents the full results of the 1-Diffractor utility tests, for all of the above-mentioned experiment parameters. Figure 2 summarizes this information by illustrating the average utility drop (in percentage points) across all GLUE tasks, for each (*mechanism*, L, ε) tuple. Similarly, Figure 3 summarizes the utility results for the comparative test against the selected previous MLDP mechanisms.

The full scores of 1-Diffractor against the five selected MLDP mechanisms on three selected GLUE tasks can be found in Table 3.

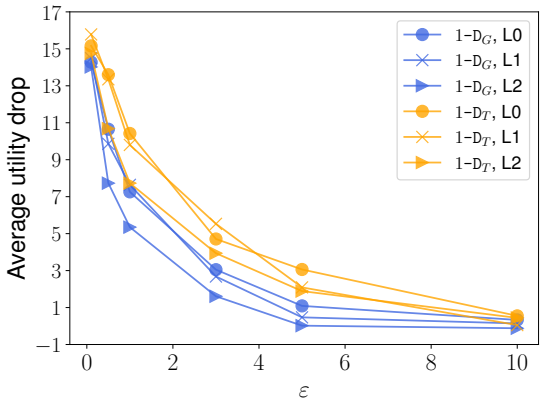


Figure 2: Average utility drop (loss) across all GLUE tasks of 1-D$_G$ and 1-D$_T$ with different list configurations and ε values. Lower scores imply higher preserved utility.

SBERT. As a final part of our conducted utility experiments, we employ an SBERT model [29], namely `all-MiniLM-L6-v2`, to compare the semantic similarity between original and privatized sentences. This approach is similar to that proposed in BERTScore [42] for evaluating text generation. As noted by Mattern et al. [23], this metric is important for evaluating the ability of a text obfuscation mechanism to preserve the semantic coherence of the original sentence. The scores, grouped by ε value, are provided in Table 2.

5 PRIVACY EXPERIMENTS

To evaluate the privacy-preservation capabilities of 1-Diffractor, we employ a two-fold approach. Firstly, the theoretical privacy guarantees of the model are tested against previous MLDP mechanisms. Next, empirical privacy tests are run on two adversarial tasks to evaluate the ability of 1-Diffractor to obfuscate text.

[1] 300-dim versions, except for E2 (200-dim)

		CoLA			SST2			QQP			MRPC			STSB (PCS)		
	Baseline	$80.57_{0.6}$			$89.98_{0.5}$			$85.24_{0.2}$			$77.78_{0.8}$			$88.17_{0.6}$		
	ε / L	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2
1-D_T	0.1	$68.49_{0.7}$	$68.01_{1.6}$	$69.00_{0.2}$	$69.30_{0.3}$	$67.16_{0.5}$	$69.50_{1.0}$	$73.50_{0.6}$	$73.40_{0.7}$	$73.50_{0.7}$	$68.55_{2.2}$	$69.85_{1.2}$	$70.02_{0.8}$	$39.40_{5.1}$	$41.42_{2.4}$	$41.95_{1.8}$
	0.5	$68.65_{0.4}$	$68.94_{0.7}$	$68.14_{1.1}$	$74.73_{1.0}$	$74.54_{1.4}$	$76.45_{0.5}$	$73.95_{1.4}$	$73.79_{1.2}$	$74.98_{0.7}$	$69.12_{2.3}$	$70.42_{0.2}$	$72.39_{0.6}$	$49.83_{2.4}$	$49.67_{0.8}$	$59.34_{0.8}$
	1	$68.36_{0.7}$	$69.16_{0.0}$	$70.63_{1.1}$	$77.06_{0.2}$	$76.99_{0.6}$	$81.27_{0.7}$	$77.24_{0.9}$	$76.57_{0.3}$	$78.27_{0.5}$	$74.75_{1.4}$	$72.55_{0.0}$	$72.96_{0.8}$	$59.38_{0.7}$	$64.77_{1.1}$	$69.38_{1.1}$
	3	$73.60_{0.5}$	$72.61_{0.8}$	$74.59_{0.2}$	$85.93_{0.7}$	$85.40_{0.3}$	$88.07_{0.7}$	$80.02_{0.6}$	$80.40_{0.3}$	$80.77_{0.5}$	$72.88_{0.9}$	$74.84_{0.7}$	$73.12_{0.8}$	$77.69_{0.4}$	$79.00_{0.8}$	$80.18_{0.8}$
	5	$75.74_{1.0}$	$77.21_{0.6}$	$76.80_{0.5}$	$88.23_{0.4}$	$88.57_{0.5}$	$89.79_{0.2}$	$82.98_{0.2}$	$82.40_{0.3}$	$83.28_{0.3}$	$75.82_{1.3}$	$75.82_{1.0}$	$75.41_{1.7}$	$83.13_{0.6}$	$83.65_{0.5}$	$84.80_{0.7}$
	10	$80.82_{0.4}$	$81.43_{0.9}$	$80.41_{0.2}$	$89.60_{0.5}$	$89.07_{0.2}$	$89.22_{0.3}$	$84.66_{0.5}$	$84.31_{0.4}$	$84.42_{0.2}$	$78.84_{0.3}$	$80.39_{0.5}$	$77.29_{0.2}$	$87.26_{0.5}$	$87.52_{0.7}$	$87.63_{0.6}$
1-D_G	0.1	$68.01_{1.2}$	$67.95_{1.5}$	$68.36_{0.5}$	$69.30_{1.3}$	$70.95_{0.5}$	$71.75_{1.0}$	$73.75_{0.5}$	$74.42_{0.5}$	$74.42_{0.5}$	$70.34_{0.5}$	$70.18_{1.2}$	$68.71_{2.0}$	$42.04_{2.4}$	$44.33_{2.1}$	$45.25_{1.8}$
	0.5	$69.10_{1.1}$	$69.22_{0.3}$	$69.42_{0.4}$	$77.10_{0.6}$	$80.43_{0.6}$	$81.46_{1.0}$	$76.73_{0.5}$	$74.92_{1.2}$	$77.06_{0.6}$	$70.59_{0.5}$	$71.57_{1.0}$	$71.08_{0.4}$	$64.06_{1.7}$	$63.40_{0.8}$	$68.60_{0.7}$
	1	$71.11_{0.3}$	$70.69_{0.8}$	$71.49_{0.3}$	$82.95_{0.5}$	$81.80_{0.1}$	$85.40_{0.8}$	$77.81_{0.7}$	$78.65_{0.1}$	$80.63_{0.1}$	$75.90_{1.5}$	$72.39_{0.6}$	$74.02_{0.5}$	$71.11_{0.7}$	$73.24_{0.4}$	$76.36_{1.2}$
	3	$77.34_{0.6}$	$77.05_{0.7}$	$77.69_{0.4}$	$88.19_{0.6}$	$89.41_{0.7}$	$89.79_{0.7}$	$82.42_{0.4}$	$82.46_{0.4}$	$83.48_{0.3}$	$75.74_{1.2}$	$74.51_{1.0}$	$77.21_{0.7}$	$84.07_{0.5}$	$83.11_{0.7}$	$84.81_{0.4}$
	5	$78.62_{0.9}$	$79.45_{0.1}$	$80.25_{0.3}$	$89.41_{0.1}$	$88.88_{1.1}$	$89.37_{0.5}$	$83.91_{0.4}$	$84.03_{0.3}$	$84.61_{0.4}$	$78.10_{1.0}$	$77.94_{0.7}$	$78.92_{0.2}$	$87.00_{0.5}$	$87.06_{0.7}$	$87.88_{0.6}$
	10	$81.08_{0.8}$	$79.71_{0.4}$	$80.98_{0.4}$	$89.56_{0.5}$	$88.99_{0.1}$	$89.72_{0.4}$	$83.79_{0.3}$	$84.18_{0.1}$	$84.88_{0.1}$	$77.37_{0.8}$	$77.86_{0.5}$	$78.68_{0.2}$	$87.07_{0.5}$	$87.53_{0.8}$	$88.14_{0.5}$

(a) Utility Scores (Accuracy) for the Classification and Textual Similarity Tasks of GLUE. Note: for STSB, the (scaled) Pearson-Spearman Correlation (PCS) is given.

		MNLI			QNLI			WNLI			RTE		
	Baseline	$76.54_{0.5}$			$84.79_{0.5}$			$38.97_{2.4}$			$59.21_{1.4}$		
	ε / L	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2
1-D_T	0.1	$54.35_{1.6}$	$54.18_{1.7}$	$54.93_{1.1}$	$67.98_{0.3}$	$67.58_{0.8}$	$69.08_{0.7}$	$48.83_{3.3}$	$41.31_{1.8}$	$46.01_{1.8}$	$54.39_{3.7}$	$56.32_{0.5}$	$54.51_{1.1}$
	0.5	$56.87_{1.3}$	$57.36_{1.1}$	$60.33_{1.0}$	$69.18_{1.0}$	$71.33_{0.5}$	$72.99_{0.8}$	$44.13_{5.8}$	$42.72_{2.9}$	$46.95_{2.4}$	$52.35_{0.6}$	$52.35_{0.8}$	$53.55_{4.7}$
	1	$61.27_{0.6}$	$60.85_{1.2}$	$63.57_{0.9}$	$73.45_{0.6}$	$72.99_{1.2}$	$76.94_{0.3}$	$42.72_{2.4}$	$44.13_{3.3}$	$42.72_{2.9}$	$53.19_{1.8}$	$54.99_{1.3}$	$55.96_{1.5}$
	3	$68.37_{1.0}$	$68.23_{0.9}$	$71.19_{0.4}$	$80.82_{0.6}$	$78.43_{0.8}$	$79.41_{0.8}$	$43.19_{1.3}$	$37.56_{3.7}$	$40.38_{2.4}$	$56.32_{0.8}$	$54.99_{0.3}$	$58.12_{0.6}$
	5	$72.77_{0.4}$	$72.78_{0.5}$	$74.12_{0.2}$	$80.37_{0.4}$	$82.31_{0.3}$	$81.16_{0.3}$	$36.15_{2.4}$	$39.44_{2.3}$	$39.91_{1.3}$	$58.48_{0.6}$	$60.29_{2.8}$	$58.97_{2.1}$
	10	$76.43_{0.3}$	$75.72_{0.6}$	$76.71_{0.3}$	$83.33_{0.0}$	$84.06_{0.2}$	$82.79_{0.6}$	$36.15_{4.4}$	$38.03_{4.1}$	$38.03_{1.1}$	$59.21_{0.8}$	$60.29_{1.0}$	$60.77_{1.6}$
1-D_G	0.1	$55.85_{1.3}$	$54.75_{1.3}$	$56.60_{1.3}$	$66.90_{0.9}$	$68.41_{0.7}$	$69.67_{1.1}$	$52.58_{2.9}$	$47.42_{7.7}$	$45.07_{1.1}$	$53.91_{3.2}$	$52.47_{3.0}$	$55.23_{1.9}$
	0.5	$59.90_{1.2}$	$61.34_{0.8}$	$63.91_{1.0}$	$72.71_{0.9}$	$71.89_{1.0}$	$74.18_{0.4}$	$41.31_{4.6}$	$40.38_{2.4}$	$46.95_{6.3}$	$53.91_{2.5}$	$59.33_{2.4}$	$58.97_{0.6}$
	1	$64.72_{1.1}$	$65.80_{0.9}$	$67.98_{0.8}$	$74.62_{1.0}$	$76.89_{1.0}$	$80.30_{0.7}$	$38.97_{0.7}$	$38.50_{2.4}$	$39.91_{4.8}$	$58.72_{1.2}$	$54.51_{1.5}$	$57.04_{2.1}$
	3	$73.47_{0.7}$	$73.83_{0.5}$	$75.05_{0.5}$	$80.61_{0.7}$	$81.76_{0.4}$	$82.37_{0.7}$	$35.21_{3.0}$	$35.21_{1.1}$	$38.03_{2.3}$	$56.80_{0.9}$	$59.81_{0.9}$	$58.36_{1.2}$
	5	$75.54_{0.5}$	$76.10_{0.4}$	$76.79_{0.6}$	$82.00_{0.7}$	$84.36_{0.3}$	$83.04_{0.7}$	$38.50_{1.3}$	$40.38_{1.3}$	$39.44_{2.0}$	$58.36_{0.2}$	$58.84_{0.5}$	$60.77_{0.9}$
	10	$76.17_{0.3}$	$76.49_{0.3}$	$77.13_{0.4}$	$85.17_{0.2}$	$85.01_{0.1}$	$84.97_{0.4}$	$38.03_{4.0}$	$38.50_{3.7}$	$38.03_{4.0}$	$60.05_{0.3}$	$61.73_{1.3}$	$59.81_{0.5}$

(b) Utility Scores (Accuracy) for the Textual Entailment Tasks of GLUE.

Table 1: Utility Scores for 1-Diffractor across the nine GLUE tasks, for six selected ε values. L values denote the three different list configuration settings that were used for the utility experiments. Scores presented are an average of three separate training runs, and the standard deviation is also presented for each score (as a subscript).

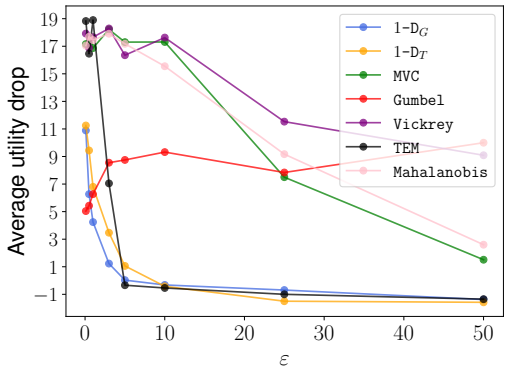


Figure 3: Average utility drop across the SST2, MRPC, and RTE tasks compared to the five selected MLDP mechanisms.

ε	0.1	0.5	1	3	5	10	25	50
$1 - D_G$	**0.37**	**0.57**	**0.66**	**0.80**	**0.82**	**0.82**	0.99	0.99
$1 - D_T$	0.31	0.48	0.57	0.73	0.79	**0.82**	**1.00**	**1.00**
MVC	0.08	0.08	0.08	0.09	0.10	0.15	0.64	0.77
Gumbel	0.33	0.33	0.33	0.33	0.33	0.33	0.59	0.61
Vickrey	0.08	0.08	0.08	0.09	0.10	0.14	0.42	0.49
TEM	0.10	0.10	0.11	0.38	0.45	0.45	0.83	0.83
Mahalanobis	0.09	0.08	0.09	0.09	0.10	0.14	0.56	0.75

Table 2: Average SBERT cosine similarity scores for (original, perturbed) sentence pairs of the MRPC, RTE, and SST2 tasks. 1-D variants are averaged for all list configurations (L0-L2). Best scores for each ε value are bolded.

5.1 Plausible Deniability

Following Feyistan et al. [14], we calculate plausible deniability statistics for our mechanism, which provide an idea of the variation introduced by a given text perturbation mechanism, for a given ε value. In particular, there are two statistics: N_w, which measures the probability that a word is returned unperturbed (i.e., is perturbed to itself), and S_w, which measures the *support* of perturbing a certain word, i.e, how many output words are expected given an input word. To estimate N_w and S_w, we randomly sample 100 words from the vocabulary of models E1-5. Using list configuration L0, each of the 100 words is perturbed 100 times through 1-Diffractor.

Figure 4 displays the results of these tests. For comparison, we also test five recent MLDP mechanisms, namely: `MVC` [14], `Gumbel`

	Mechanism / ε	0.1	0.5	1	3	5	10	25	50
SST2	1-D Best	71.75 ± 1.0	**81.46 ± 1.0**	**85.40 ± 0.8**	**89.79 ± 0.7**	**89.79 ± 0.2**	**89.72 ± 0.4**	89.72 ± 0.6	89.72 ± 0.6
	MVC	53.78 ± 2.0	55.01 ± 3.0	52.98 ± 1.5	54.32 ± 2.4	57.42 ± 1.2	59.52 ± 0.8	80.05 ± 0.8	86.23 ± 0.8
	Gumbel	**78.44 ± 0.4**	77.06 ± 0.1	75.27 ± 1.4	76.72 ± 1.4	76.61 ± 1.9	76.61 ± 1.2	78.36 ± 1.4	76.41 ± 0.2
	Vickrey	53.25 ± 1.7	55.16 ± 0.7	53.90 ± 2.1	53.59 ± 2.0	56.96 ± 0.8	56.88 ± 0.8	71.67 ± 0.5	74.16 ± 1.0
	TEM	53.13 ± 0.8	54.97 ± 2.9	51.34 ± 0.7	78.10 ± 0.4	89.26 ± 0.4	89.14 ± 1.5	**90.14 ± 0.5**	**90.21 ± 1.1**
	Mahalanobis	54.74 ± 3.0	55.01 ± 2.4	54.01 ± 2.2	53.78 ± 1.5	53.82 ± 1.4	59.40 ± 0.7	74.89 ± 0.4	85.32 ± 0.4
RTE	1-D Best	**56.32 ± 0.5**	**59.33 ± 2.4**	**58.72 ± 1.2**	**59.81 ± 0.9**	**60.77 ± 0.9**	**61.73 ± 1.3**	**65.00 ± 3.6**	**63.70 ± 2.7**
	MVC	52.59 ± 1.0	52.47 ± 1.6	54.51 ± 1.0	50.42 ± 2.5	48.38 ± 1.1	48.86 ± 1.9	55.20 ± 0.0	59.81 ± 3.3
	Gumbel	**56.32 ± 2.3**	54.39 ± 0.3	54.99 ± 2.2	53.67 ± 1.6	52.95 ± 1.2	52.71 ± 2.1	54.99 ± 0.7	51.26 ± 1.8
	Vickrey	51.99 ± 0.6	54.15 ± 2.5	51.62 ± 3.1	51.26 ± 2.7	52.59 ± 0.5	49.22 ± 1.5	53.79 ± 0.8	55.60 ± 0.3
	TEM	49.94 ± 2.7	53.19 ± 2.2	51.87 ± 3.4	58.84 ± 1.9	59.81 ± 1.0	60.29 ± 1.4	61.50 ± 3.3	62.21 ± 3.8
	Mahalanobis	53.43 ± 1.5	51.50 ± 0.9	51.99 ± 1.2	52.23 ± 1.2	51.99 ± 1.6	53.43 ± 1.8	54.63 ± 2.0	59.57 ± 1.3
MRPC	1-D Best	70.34 ± 0.5	**72.39 ± 0.6**	**75.90 ± 1.5**	**77.21 ± 0.7**	**78.92 ± 0.2**	**80.39 ± 0.5**	**79.24 ± 0.7**	**79.41 ± 0.9**
	MVC	69.12 ± 0.6	67.89 ± 1.0	68.87 ± 0.3	67.57 ± 1.3	69.28 ± 0.4	66.67 ± 1.7	69.20 ± 0.7	76.39 ± 0.9
	Gumbel	**70.59 ± 0.9**	71.73 ± 0.5	69.44 ± 1.5	70.92 ± 0.6	71.16 ± 0.8	69.69 ± 1.4	70.10 ± 0.6	69.28 ± 0.2
	Vickrey	67.97 ± 3.2	66.99 ± 1.3	68.46 ± 0.5	67.24 ± 0.8	68.38 ± 1.1	67.97 ± 0.6	66.91 ± 1.0	69.93 ± 0.7
	TEM	67.40 ± 2.3	69.44 ± 0.8	67.08 ± 2.7	68.87 ± 3.7	**78.92 ± 0.5**	79.17 ± 0.3	78.35 ± 0.6	78.59 ± 0.3
	Mahalanobis	67.65 ± 3.0	67.40 ± 3.0	68.63 ± 0.9	67.24 ± 2.0	69.61 ± 1.0	67.48 ± 0.7	69.93 ± 0.5	74.26 ± 1.2

Table 3: Utility Experiment Results with previous MLDP mechanisms, on three selected GLUE tasks. Scores represent the average of three runs, and standard deviations are presented. *1-D Best* denotes the highest score achieved by a 1-Diffractor configuration, i.e., (*mechanism*, L) pair, from all scores presented in Table 1. Bolded values represent the best scores per ε value.

[37], `Vickrey` [39], `TEM` [5], and `Mahalanobis` [38], the set of mechanisms used for comparative testing in the remainder of this work.

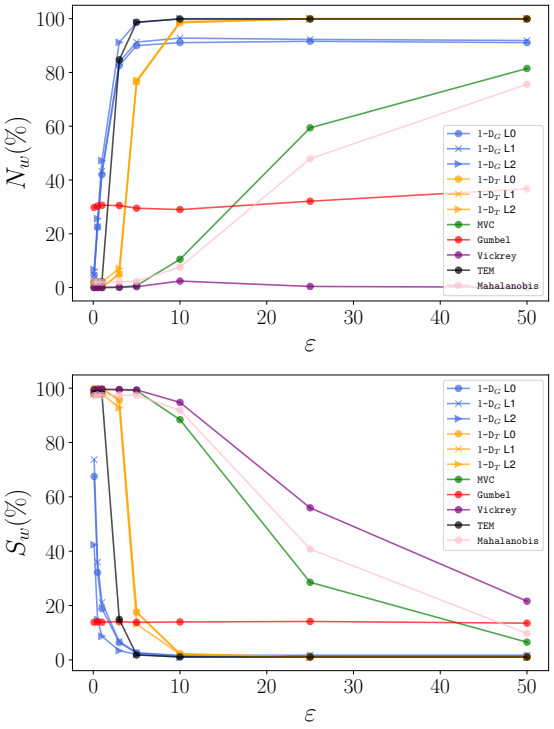


Figure 4: Empirical N_w and S_w statistics for 1-Diffractor and five selected MLDP mechanisms.

5.2 Empirical Privacy

For empirical privacy tests, we choose two tasks: *speaker identification* and *gender identification*. The results are visualized in Figure 5 and presented in full in Table 4.

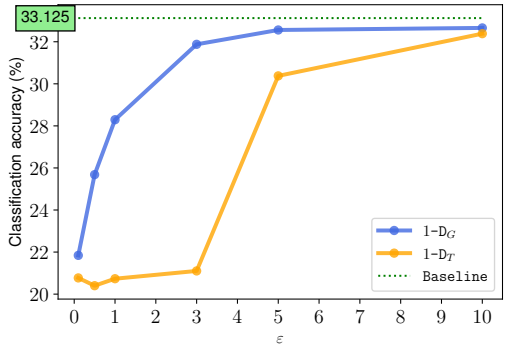
In the speaker identification task (**FI**), we use the *Friends Corpus* [8], which contains the entirety of the script from the TV show *Friends*. We take a subset of only the six main characters, and fine-tune a BERT model to identify a character based on their line. In the adversarial setting, this model mimics an attacker who wishes to identify authors based upon publicly accessible textual data.

In the second task (**TG**), we use a dataset of US-based reviews on *Trustpilot* [17]. Each review has been marked with the gender of the author. From this, we fine-tune BERT to predict the gender of an author based on the review text.

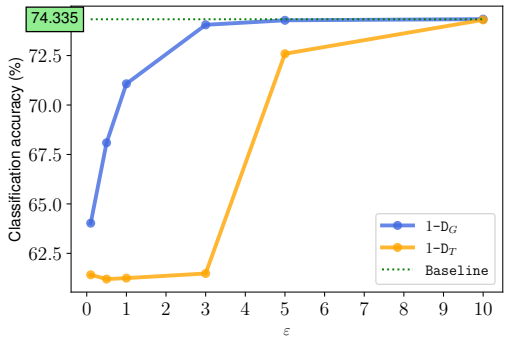
In both cases, the model acting as our adversary is trained with an 80% split of the dataset, using a 10% validation set. The 10% test set is used to obtain the baseline scores for the adversarial classifier. Next, the test set is perturbed using 1-Diffractor with the mentioned ε values, using the **L0** configuration. Finally, the adversarial accuracy is evaluated for each perturbed dataset.

	ε	0.1	0.5	1	3	5	10
FI	*Baseline*	33.13					
	1-D$_G$	21.84	25.68	28.29	31.88	32.56	32.66
	1-D$_T$	20.77	20.40	20.73	21.10	30.38	32.39
TG	*Baseline*	74.34					
	1-D$_G$	64.03	68.09	71.07	74.06	74.28	74.34
	1-D$_T$	61.42	61.20	61.25	61.48	72.59	74.31

Table 4: Complete empirical privacy results (accuracy). *FI* = Friends identification task, *TG* = Trustpilot gender task.



(a) FI results.



(b) TG results.

Figure 5: Emprical Privacy Results. *FI* = Friends identification task, *TG* = Trustpilot gender task.

5.3 A Note on Epsilon

In the comparative analysis presented, it is important to note that the choice of ε does not necessarily equate to the same privacy guarantees across mechanisms. This is due to different distance metrics being used, which scale the chosen ε according to MDP. Therefore, the ε values in our experiments are chosen for comparison, not necessarily to equate the *effective* ε values. Nevertheless, we mitigate this challenge by also testing all mechanisms including 1-DIFFRACTOR on $\varepsilon \in \{25, 50\}$ in the comparative evaluations, thereby extending the investigated range. These results can be found in Tables 2 and 3, as well as in Figures 3 and 4.

6 EFFICIENCY EXPERIMENTS

The final experiments aim to measure the scalability of our proposed 1-DIFFRACTOR mechanism, in comparison to previous MLDP mechanisms. This is performed by measuring the *speed* and *memory consumption* of each mechanism, quantified by the number of tokens that can be perturbed in a day and memory usage per word perturbation, respectively. Note that in these calculations, the list initialization of 1-DIFFRACTOR is not included, as the time is negligible (ca. 20 seconds per list on a CPU) with respect to the 24-hour period, and 150 MiB for the initialization of **L0**.

We first estimate efficiency by capturing the amount of time it takes to perturb a random set of 1000 words from the list vocabulary. Next, we measure efficiency empirically by using each mechanism to perturb the complete SST2 dataset. The number of perturbed tokens is divided by the elapsed and then extrapolated to 24 hours.

The second set of experiments focuses on the memory consumption of the word perturbations. To measure this, we measure the memory needed to perturb the same set of 1000 words as introduced above, using the Python `memory-profiler` package.

The results are summarized in Figures 6-7 and Table 5. All experiments were run on a single 8-core Intel Xeon 2.20 GHz CPU.

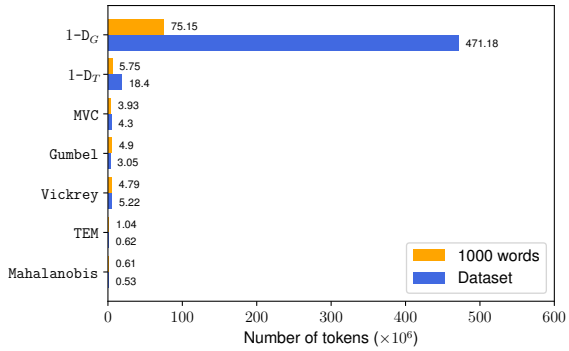


Figure 6: Comparison of number of tokens that can be processed per day by calculating the speed over 1000 words and extrapolating to 24 hours (denoted by *1000 words*). *Dataset* is derived from the perturbation of the SST2 dataset.

	1-D$_G$	1-D$_T$	MVC	Gumbel	Vickrey	TEM	Mahalanobis
total	0.05	0.01	206.89	118.96	170.44	81.34	78.08
per-word	0.00005	0.00001	0.207	0.119	0.170	0.081	0.078

Table 5: Memory consumption (in MiB) for 1000 words perturbed. Note that for both 1-DIFFRACTOR settings, the consumption does not include the initial list configuration, as this is a one-time cost, and does not accumulate per word.

7 DISCUSSION

Analysis of Experiments. Leading the discussion on the merits of 1-DIFFRACTOR, analysis begins with the basis of our one-dimensional word lists at the basis of the mechanism. The effect can be seen in Figure 4, where including double the number of lists (**L1**) provides a slight increase in privacy (plausible deniability) over only one list per model (**L0**). This comes with a negligible difference in utility loss (Figure 2), thus prompting further investigation into the usage of an even greater number of lists simultaneously.

With the use of only *1 model, 1 list* (**L2**) in the case of both 1-D$_G$ and 1-D$_T$, the **L2** configuration provides clearly better utility preservation, at the cost of lower privacy protection, as shown in Figure 4. Thus, one can begin to observe the notion that while a greater number of (embedding) lists may increase empirical privacy, a carefully selected single list may be best for utility preservation.

Looking to the impact of ε, the utility performance across ε values exhibit a clear "privacy-utility trade-off curve", with an average utility drop of around 15 percentage points at 0.1, and near baseline performance for values above 5. The benefit of 1-DIFFRACTOR is

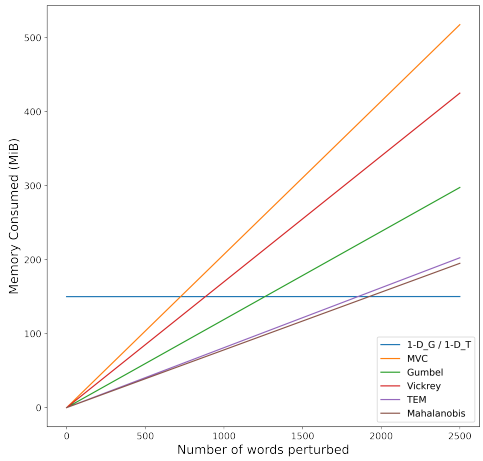


Figure 7: Memory consumption of all compared mechanisms as a function of words perturbed, using the per-word rates of Table 5. For the 1-Diffractor mechanisms, the memory required for initial list configuration is accounted for (see Section 6). In addition, the two 1-Diffractor variants are plotted on the same line, due to their negligibly small difference in memory required. Note that these plots assume a linear growth in memory consumption over time.

made salient by these results: within a relatively small range of ε values, one can observe high levels of perturbation (measured by N_w) and considerably lower utility scores, and vice versa. The effectiveness of this obfuscation is solidified by the empirical privacy results, showing that it reduces the predictive power of adversaries.

As opposed to other tested mechanisms, 1-Diffractor exhibits a much "tighter" ε region, made most clear by Figures 4 and 5. This can be attributed to the reduction to one dimension, where the overall magnitude of noise added is less as compared to MLDP mechanisms operating on distance metrics in higher dimensions. The simplicity of 1-Diffractor thus leads to much smaller, interpretable privacy budgets, which follow a "graceful" degradation in privacy (and increase in utility) as the ε value increases. While one may argue that this is simply a matter of scale, the benefit of such a bounded region can be interpreted as a clearer range of acceptable privacy budgets. This would certainly be necessary for adoption into practice.

The Privacy-Efficiency Trade-off. The design of MLDP mechanisms often is evaluated for utility and privacy, sparking debates on the privacy-utility trade-off, yet the literature has largely ignored the question of efficiency, which would reasonably be required for such mechanisms to be employed in practice and at scale.

Our results show that 1-Diffractor greatly outperforms previous methods in terms of the speed at which words are perturbed. This is particularly the case with our 1-D_G variant, which achieves immense speedups over previous MLDP mechanisms: over 15x from a theoretical estimate and over 90x from an empirical measurement (see Figure 6). Furthermore, these speedups can be realized on everyday hardware, i.e., a standard laptop CPU.

Such a speedup of course comes with a trade-off. As shown in our privacy experiments, 1-D_G exhibits lower theoretical privacy guarantees (via plausible deniability statistics) and demonstrates less effectiveness at lowering adversarial advantage. This is placed in juxtaposition to 1-D_T, which operates at a considerably slower rate, yet demonstrates higher privacy benchmarks.

In the memory consumption benchmarks, 1-Diffractor also shows significant improvements over previous mechanisms, even when accounting for the memory required to initialize the word lists (see Figure 7). This can be attributed to the fact that post-initialization, 1-Diffractor does not need to perform expensive nearest neighbor searches for each word, which must be done in all other compared mechanisms. The resulting difference can be clearly observed in the per-word rates of Table 5. As can be seen in Figure 7, after only 2000 words perturbed, 1-Diffractor already begins to use less memory than all other mechanisms.

The Question of an Optimal Obfuscation Mechanism. In analyzing 1-Diffractor, it is clear that even at lower ε values, not as many tokens will be perturbed away from the original token. For example, at $\varepsilon = 1$, only around 60% of tokens are perturbed, as per the N_w statistic, which is still significantly lower than other mechanisms even at higher values such as 10 (e.g., Mahalanobis $N_w@10 = 0.92$).

The debate here becomes how to interpret a "good" obfuscation mechanism. If nearly 100% tokens are perturbed, this may grant high plausible deniability, but the utility will be impacted significantly. This is especially true in the case of high-dimensional perturbations, where privatized words may be far in meaning from the original word due to the effect of noise across many dimensions.

Mattern et al. [23] echo the need for balanced obfuscation, namely not only with a high perturbation rate, but also in the preservation of semantic meaning and relatedly, utility in downstream tasks and empirically demonstrable privacy. In this work, we add to this notion, arguing that efficiency (speed) is also a key factor.

With 1-Diffractor, we demonstrate such a balance, as the mechanism successfully preserves privacy via word perturbations, measured via plausible deniability and empirical privacy, while still allowing for the utility to remain intact. This is due not only to *how often* words are perturbed, but also *how* words are perturbed.

Open Challenges. Our evaluation of 1-Diffractor verifies its utility- and privacy-preserving capabilities, as well as a significant speedup over previous methods, yet a discussion of its merits must be accompanied by an analysis of open questions.

A major limitation of word-level MLDP methods comes with the inability to preserve grammatically correct sentence structures, due to the lack of context in single-word perturbations. To a degree, 1-Diffractor preserves sentence coherence in the way that the lists are built: similar words functionally will (ideally) be sorted near each other. Nevertheless, this does not always hold, particularly at smaller ε values.The strength of 1-Diffractor over previous MLDP mechanisms, however, is clearly demonstrated in our results. As an added limitation due to the word-level nature, our method cannot construct obfuscated outputs of differing length from the original sentence, another issue highlighted by Mattern et al. [23].

In continuing the discussion of Section 5.3, it is important to keep in mind the limitation of interpreting ε across different MLDP mechanisms. The variation of underlying metrics makes evaluation challenging, as there exists no standard way of evaluating

cross-metric DP mechanisms. Nevertheless, we address this shortcoming by testing on a wide range of ε values. In this, we show that 1-DIFFRACTOR remains competitive across all tested values, albeit with a diminishing theoretical privacy guarantee at higher values (i.e., 25 and 50). A concrete improvement is shown with 1-DIFFRACTOR, in both versions, against the directly comparable TEM [5], where our method consistently achieves higher utility scores while maintaining similar privacy levels. As mentioned above, this of course comes with the added benefit of higher perturbation speeds regardless of ε value. Comparing once again to the original TEM of Carvalho et al. [5], one can observe in Figure 6 the effects of the dimensionality reduction offered by our list structure, where our 1-D$_T$ performs significantly faster than the original TEM.

A final point comes with the question of dataset and task dependence. In our tests, we focus on a variety of tasks, from sentence similarity to textual entailment, yet we do not study the effect of each specific task. For example, one can see from Table 1 that the entailment tasks are most affected in terms of utility (see MNLI and QNLI). In addition, regression tasks (i.e., STSB) are also affected more severely as opposed to classification. Factors like these call for more in-depth analyses of MLDP mechanism design and evaluation.

8 RELATED WORK

The notion of *word-level* Metric (Local) DP was introduced by Fernandes et al. [12]. This inspired several follow-up works [5, 14, 15, 37–39], which investigate the usage of various noise mechanisms or metric spaces. These works rely on embedding perturbations, which are carried out with DP to achieve private embeddings [16, 20].

At the same time, other earlier works diverged from the word-level MLDP notion, focusing instead on private model training [1], differentially private word replacement selection [7, 40] or privacy-preserving neural representations of text [2, 22]. A critique of earlier methods, particularly at the word level, by Mattern et al. [23] highlights several shortcomings, as noted previously in this work. Other works [13, 20] echo some of these challenges.

In light of these limitations, further works investigate the integration of DP in more advanced NLP models, such as earlier works on encoder(-decoder) models [3, 28]. Other works extend beyond the word level to the sentence and document level [24]. In recent state-of-the-art approaches, DP is achieved in combination with the training and fine-tuning of language models [10, 30], or in directly adding noise to the latent representation, such as in DP-BART [18].

Recent methods address the issue of DP-rewritten sentences with grammatical correctness [28, 33, 35, 41]. However, such methods rely on the utilization of computationally expensive language models, thus lacking scalability. Other methods, such as DP-BART [18], rely on noise addition in high dimensions, thus leading to very large privacy budgets. Here, the "individual" more complex, as opposed to a word vocabulary with discrete and finite members.

Building upon these previous works, we follow in the footsteps of existing word-level MLDP mechanisms, focusing on efficiency while avoiding the usage of computationally expensive language models. In this way, we hope to advance the field of text privatization by emphasizing the design of utility- and privacy-preserving mechanisms that are lightweight and accessible to run.

9 CONCLUSION

In this work, we introduce 1-DIFFRACTOR, a novel word-level MLDP mechanism for text obfuscation. 1-DIFFRACTOR is built upon a simple and intuitive method of sorting words in one-dimensional lists, which serve as the basis for word privatization via Metric DP, achieved through a *diffraction* of noise along this dimension. In a three-part evaluation, our method exhibits utility- and privacy-preserving capabilities, while notably demonstrating significant efficiency improvements over previous MLDP mechanisms.

Our findings illustrate the merit of researching novel ways of representation for text privatization, showcasing that word-level perturbations are effective on the utility and privacy fronts, while also possessing the ability to be deployed at scale. This lightweightness makes a salient case for further research and future improvements.

We see three paths of further research to build upon our work, as well as its perceived limitations: (1) exploration into the effect of different word embedding models, as well as their combination and the use of multiple lists, (2) work on the creation of a uniform benchmark for word-level MLDP mechanisms, regardless of the underlying metric, and (3) relatedly, in-depth research into the design and implementation of utility **and** privacy-preserving mechanisms, that also can be practically deployed at scale.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Ghazaleh Beigi, Kai Shu, Ruocheng Guo, Suhang Wang, and Huan Liu. 2019. Privacy Preserving Text Representation Learning. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media* (Hof, Germany) *(HT '19)*. Association for Computing Machinery, New York, NY, USA, 275–276. https://doi.org/10.1145/3342220.3344925

[3] Haohan Bo, Steven H. H. Ding, Benjamin C. M. Fung, and Farkhund Iqbal. 2021. ER-AE: Differentially Private Text Generation for Authorship Anonymization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 3997–4007. https://doi.org/10.18653/v1/2021.naacl-main.314

[4] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[5] Ricardo Silva Carvalho, Theodore Vasiloudis, Oluwaseyi Feyisetan, and Ke Wang. 2023. TEM: High utility metric differential privacy on text. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 883–890. https://doi.org/10.1137/1.9781611977653.ch99

[6] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the scope of differential privacy using metrics. In *Privacy enhancing technologies (Lecture notes in computer science)*, Emiliano De Cristofaro and Matthew Wright (Eds.). Springer, Springer Nature, United States, 82–102. https://doi.org/10.1007/978-3-642-39077-7_5 International Symposium on Privacy Enhancing Technologies (13th : 2013) ; Conference date: 10-07-2013 Through 12-07-2013.

[7] Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. 2023. A Customized Text Sanitization Mechanism with Differential Privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5747–5758. https://doi.org/10.18653/v1/2023.findings-acl.355

[8] Yu-Hsin Chen and Jinho D. Choi. 2016. Character Identification on Multiparty Conversation: Identifying Mentions of Characters in TV Shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Los Angeles, 90–100. https://doi.org/

10.18653/v1/W16-3612

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Minxin Du, Xiang Yue, Sherman S. M. Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. DP-Forward: Fine-Tuning and Inference on Language Models with Differential Privacy in Forward Pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (<conf-loc>, <city>Copenhagen</city>, <country>Denmark</country>, </conf-loc>) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2665–2679. https://doi.org/10.1145/3576915.3616592

[11] Cynthia Dwork. 2006. Differential Privacy. In *International Colloquium on Automata, Languages and Programming*. https://api.semanticscholar.org/CorpusID:2565493

[12] Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. In *Principles of Security and Trust: 8th International Conference, POST 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings 8*. Springer International Publishing, 123–148. https://doi.org/10.1007/978-3-030-17138-4_6

[13] Oluwaseyi Feyisetan, Abhinav Aggarwal, Zekun Xu, and Nathanael Teissier. 2021. Research Challenges in Designing Differentially Private Text Generation Mechanisms. In *The International FLAIRS Conference Proceedings*, Vol. 34. https://doi.org/10.32473/flairs.v34i1.128461

[14] Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and Utility-Preserving Textual Analysis via Calibrated Multivariate Perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 178–186. https://doi.org/10.1145/3336191.3371856

[15] Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. 2019. Leveraging hierarchical representations for preserving privacy and utility in text. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 210–219. https://doi.org/10.1109/ICDM.2019.00031

[16] Oluwaseyi Feyisetan and Shiva Kasiviswanathan. 2021. Private Release of Text Embedding Vectors. In *Proceedings of the First Workshop on Trustworthy Natural Language Processing*. Association for Computational Linguistics, Online, 15–27. https://doi.org/10.18653/v1/2021.trustnlp-1.3

[17] Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User Review Sites as a Resource for Large-Scale Sociolinguistic Studies. In *Proceedings of the 24th International Conference on World Wide Web* (Florence, Italy) *(WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 452–461. https://doi.org/10.1145/2736277.2741141

[18] Timour Igamberdiev and Ivan Habernal. 2023. DP-BART for Privatized Text Rewriting under Local Differential Privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 13914–13934. https://doi.org/10.18653/v1/2023.findings-acl.874

[19] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826. https://doi.org/10.1137/090756090

[20] Oleksandra Klymenko, Stephen Meisenbacher, and Florian Matthes. 2022. Differential Privacy in Natural Language Processing The Story So Far. In *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing*. Association for Computational Linguistics, Seattle, United States, 1–11. https://doi.org/10.18653/v1/2022.privatenlp-1.1

[21] Fragkiskos Koufogiannis, Shuo Han, and George J. Pappas. 2017. Gradual Release of Sensitive Data under Differential Privacy. *Journal of Privacy and Confidentiality* 7, 2 (Jan. 2017). https://doi.org/10.29012/jpc.v7i2.649

[22] Lingjuan Lyu, Xuanli He, and Yitong Li. 2020. Differentially Private Representation for NLP: Formal Guarantee and An Empirical Study on Privacy and Fairness. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 2355–2365. https://doi.org/10.18653/v1/2020.findings-emnlp.213

[23] Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. 2022. The Limits of Word Level Differential Privacy. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, Seattle, United States, 867–881. https://doi.org/10.18653/v1/2022.findings-naacl.65

[24] Casey Meehan, Khalil Mrini, and Kamalika Chaudhuri. 2022. Sentence-level Privacy for Document Embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3367–3380. https://doi.org/10.18653/v1/2022.acl-long.238

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013). https://doi.org/10.48550/arXiv.1301.3781

[26] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy Risks of General-Purpose Language Models. In *2020 IEEE Symposium on Security and Privacy (SP)*. 1314–1331. https://doi.org/10.1109/SP40000.2020.00095

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[28] Natalia Ponomareva, Jasmijn Bastings, and Sergei Vassilvitskii. 2022. Training Text-to-Text Transformers with Privacy Guarantees. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 2182–2193. https://doi.org/10.18653/v1/2022.findings-acl.171

[29] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[30] Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. 2022. Selective Differential Privacy for Language Modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 2848–2859. https://doi.org/10.18653/v1/2022.naacl-main.205

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

[32] Jingye Tang, Tianqing Zhu, Ping Xiong, Yu Wang, and Wei Ren. 2020. Privacy and Utility Trade-Off for Textual Analysis via Calibrated Multivariate Perturbations. In *Network and System Security*, Mirosław Kutyłowski, Jun Zhang, and Chao Chen (Eds.). Springer International Publishing, Cham, 342–353. https://doi.org/10.1007/978-3-030-65745-1_20

[33] Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally Differentially Private Document Generation Using Zero Shot Prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 8442–8457. https://doi.org/10.18653/v1/2023.findings-emnlp.566

[34] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Brussels, Belgium, 353–355. https://doi.org/10.18653/v1/W18-5446

[35] Yuyang Wang, Xianjia Meng, and Ximeng Liu. 2023. Differentially Private Recurrent Variational Autoencoder For Text Privacy Preservation. *Mobile Networks and Applications* (2023), 1–16. https://doi.org/10.1007/s11036-023-02096-9

[36] Benjamin Weggenmann and Florian Kerschbaum. 2018. SynTF: Synthetic and differentially private term frequency vectors for privacy-preserving text mining. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 305–314. https://doi.org/10.1145/3209978.3210008

[37] Nan Xu, Oluwaseyi Feyisetan, Abhinav Aggarwal, Zekun Xu, and Nathanael Teissier. 2021. Density-aware differentially private textual perturbations using truncated Gumbel noise. In *The International FLAIRS Conference Proceedings*, Vol. 34. https://doi.org/10.32473/flairs.v34i1.128463

[38] 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*. 7–17. https://doi.org/10.18653/v1/2020.privatenlp-1.2

[39] Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2021. On a Utilitarian Approach to Privacy Preserving Text Generation. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*. 11–20. https://doi.org/10.18653/v1/2021.privatenlp-1.2

[40] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. 2021. Differential Privacy for Text Analytics via Natural Text Sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 3853–3866. https://doi.org/10.18653/v1/2021.findings-acl.337

[41] Xiang Yue, Huseyin Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. 2023. Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 1321–1342. https://doi.org/10.18653/v1/2023.acl-long.74

[42] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkeHuCVFDr