Differential Privacy for Text Analytics via Natural Text Sanitization[†]

Xiang Yue[1,‡], Minxin Du[2], Tianhao Wang[3], Yaliang Li[4], Huan Sun[1], and Sherman S. M. Chow[2]

[1]The Ohio State University
[2]The Chinese University of Hong Kong
[3]Carnegie Mellon University
[4]Alibaba Group
{yue.149,sun.397}@osu.edu,{dm018,sherman}@ie.cuhk.edu.hk
tianhao@cmu.edu, yaliang.li@alibaba-inc.com

Abstract

Texts convey sophisticated knowledge. However, texts also convey sensitive information. Despite the success of general-purpose language models and domain-specific mechanisms with differential privacy (DP), existing text sanitization mechanisms still provide low utility, as cursed by the high-dimensional text representation. The companion issue of utilizing sanitized texts for downstream analytics is also under-explored. This paper takes a direct approach to text sanitization. Our insight is to consider both sensitivity and similarity via our new local DP notion. The sanitized texts also contribute to our sanitization-aware pretraining and fine-tuning, enabling privacy-preserving natural language processing over the BERT language model with promising utility. Surprisingly, the high utility does not boost up the success rate of inference attacks.

1 Introduction

Natural language processing (NLP) requires a lot of training data, which can be sensitive. Naïve redaction approaches (*e.g.*, removing common personally identifiable information) is known to fail (Sweeney, 2015): innocuous-looking fields can be linked to other information sources for reidentification. The recent success of many language models (LMs) has motivated security researchers to devise advanced privacy attacks. Carlini et al. (2020b) recover texts from (a single document of) the *training data* via querying to an LM pretrained from it. Pan et al. (2020) and Song and Raghunathan (2020) target the text embedding, *e.g.*, revealing from an encoded *query* to an NLP service.

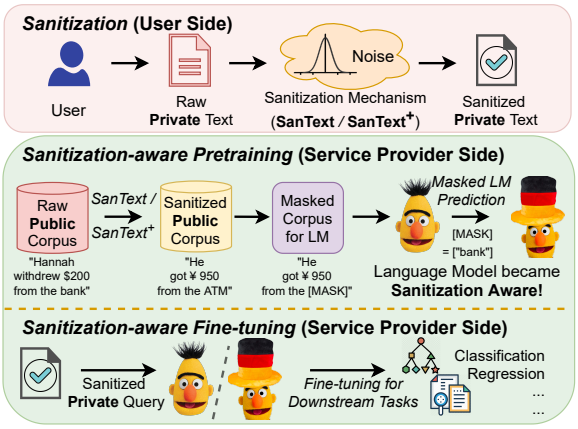

Figure 1: Workflow of our PPNLP pipeline, including the user-side sanitization and the service provider-side NLP modeling with pretraining/fine-tuning

Emerging NLP works focus on only specific document-level (statistical) features (Weggenmann and Kerschbaum, 2018) or producing private text representations (Xie et al., 2017; Coavoux et al., 2018; Elazar and Goldberg, 2018; Li et al., 2018) as initial solutions to the first issue above on training-data privacy. However, the learned representations are *not* human-readable, which makes *transparency* (*e.g.*, required by GDPR) questionable: an average user may not have the technical know-how to verify whether sensitive attributes have been removed or not. Moreover, consider the whole NLP pipeline, the learned representations often entail extra modeling or non-trivial changes to existing NLP models, which take dedicated engineering efforts.

1.1 Sanitizing Sensitive Texts, Naturally

With this state-of-affairs of the security and the NLP research, we deem it better to address privacy from the root, *i.e.*, directly producing sanitized text documents. Being the most native format, they

[†]Our code is available at https://github.com/xiangyue9607/SanText.

[‡]The first two authors contributed equally.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3853–3866

incur minimal changes to existing NLP pipelines. Being human-readable, they provide transparency (to privacy-concerning training-data contributors) and explainability (*e.g.*, to linguists who might find the need for investigating how the training data contribute to a certain result). Moreover, it naturally extends the privacy protection to the inference phase. Users can apply our sanitization mechanism before sending queries (*e.g.*, medical history) to the NLP service provider (*e.g.*, diagnosis services).

Conceptually, we take a natural approach – we sanitize text documents into also (sanitized) text documents. This is in great contrast to the typical "post-processing" for injecting noises either to gradients in training (a deep neural network) (McMahan et al., 2018) or the "cursed" *high-dimensional* text representations (Lyu et al., 2020a,b; Feyisetan et al., 2020). It also leads to our $O(1)$ efficiency, freeing us from re-synthesizing the document word-by-word via nearest neighbor searches over the entire vocabulary space $\mathcal{V}$ (Feyisetan et al., 2020).

Technically, we aim for the *de facto* standard of local differential privacy (LDP) (Duchi et al., 2013) to sanitize the user data *locally*, based on which the service provider can build NLP models without touching any raw data. DP has been successful in many contexts, *e.g.*, location privacy and survey statistics (Andrés et al., 2013; Murakami and Kawamoto, 2019). However, DP text analytics appears to be a difficult pursuit (as discussed, also see Section 2), which probably explains why there are only a few works in DP-based text sanitization. In high-level terms, text is rich in semantics, differentiating it from other more structured data.

Our challenge here is to develop *efficient* and effective mechanisms that *preserve the utility* of the text data with *provable and quantifiable* privacy guarantees. Our insight is the formulation of a new LDP notion named *Utility-optimized Metric LDP (UMLDP)*. We attribute our success to the focus of UMLDP on protecting what matters (sensitive words) via "sacrificing" the privacy of non-sensitive (common) words. To achieve UMLDP, our mechanism directly samples noises on tokens.

Our result in this regard is already better than the state-of-the-art LDP solution producing sanitized documents (Feyisetan et al., 2020) – we got 28% gain in accuracy on the SST-2 dataset (Wang et al., 2019) on average at the same privacy level (*i.e.*, the same LDP parameter) while being much more efficient ($\sim$60$\times$ faster, precomputation included).

1.2 Privacy-Preserving NLP, Holistically

Text sanitization is essential but just one piece of the whole privacy-preserving NLP (PPNLP) pipeline. While most prior works in text privacy are motivated by producing useful data for some downstream tasks, the actual text analytics are hardly explored, not to say in the context of many recent general-purpose language models. As simple as it might seem, we start to see design choices that can be influential. Specifically, our challenge here is to adapt the currently dominating pretraining-fine-tuning paradigm (*e.g.*, BERT (Devlin et al., 2019)) over sanitized texts for building the model.

Our design is to build in privacy at the root again, in contrast to the afterthought approach. We found it beneficial to sanitize even the public data before feeding them to training. It is not for protecting the public data per se. The intuition here is that it "prepares" the model to work with sanitized queries, which explains our eventual (slight) increase in accuracy while *additionally ensuring privacy*.

Specifically, we propose a sanitization-aware pretraining procedure (Figure 1). We first use our mechanisms to sanitize the public texts, mask the sanitized texts (as in BERT), and train the LM by predicting a `MASK` position as its *original unsanitized token*. LMs preptrained with our sanitization-aware procedure are expected to be more robust to noises in the sanitized texts and achieve better utility when fine-tuning on downstream tasks.

We conduct experiments on three representative NLP tasks to empirically confirm that our proposed PPNLP pipeline preserves both utility and privacy. It turns out that our sanitization-based pretraining (using only $1/6$ of data used in the original BERT pretraining) can even improve the utility of NLP tasks while maintaining privacy comparable to the original BERT. Note that there is an inherent tension between utility and privacy, and privacy attack is also inference in nature. To empirically demonstrate the privacy aspect of our pipeline, *i.e.*, it does not make our model a more powerful tool helping the attacker, we also conduct the "mask token inference" attack on private texts, which infers the masked token given its context based on BERT. As a highlight, our base solution SANTEXT improves the defense rate by 20% with only a 4% utility loss on the SST-2 dataset. We attribute our surprising result of mostly helping only good guys to our natural approach: to avoid the model memorizing sensitive texts "too well," we fed it with sanitized text.

2 Related Work

Privacy risks in NLP. A taxonomy of attacks that recover sensitive attributes or partial raw text from text embeddings output by popular LMs has been proposed (Song and Raghunathan, 2020), without any assumptions on the structures or patterns in input text. Carlini et al. (2020b) also show a powerful black-box attack on GPT-2 (Radford et al., 2019) that extracts verbatim texts of training data. Defense with rigorous guarantees (DP) is thus vital.

Differential privacy and its application in NLP. DP (Dwork, 2006) has emerged as the *de facto* standard for statistical analytics (Wang et al., 2017, 2018; Cormode et al., 2018). A few efforts inject high-dimensional DP noise into text representations (Feyisetan et al., 2019, 2020; Lyu et al., 2020a,b). The noisy representations are not human-readable and not directly usable by existing NLP pipelines, *i.e.*, they consider a different problem not directly comparable to ours. More importantly, they fail to strike a nice privacy-utility balance due to "the curse of dimensionality," *i.e.*, the magnitude of the noise is too large for high-dimensional token embedding, and thus it becomes exponentially less likely to find a noisy embedding close to a real one on every dimension. This may also explain why an earlier work focuses on document-level statistics only, *e.g.*, term-frequency vectors (Weggenmann and Kerschbaum, 2018).

Our approaches produce natively usable sanitized texts via directly sampling a substitution for each token from a precomputed distribution (to be detailed in Section 4), circumventing the dimension curse and striking a privacy-utility trade-off while being much more efficient. A concurrent work (Qu et al., 2021) also considers the whole NLP pipeline, but it still builds on the token-projection approach (Feyisetan et al., 2020).

Privacy-preserving text representations. Learning private text representations via adversarial training is also an active area (Xie et al., 2017; Coavoux et al., 2018; Elazar and Goldberg, 2018; Li et al., 2018). An adversary is trained to infer sensitive information jointly with the main model, while the main model is trained to maximize the adversary's loss and minimize the primary learning objective. While we share the same general goal, our aim is not such representations (similar to those with DP) but to release sanitized text for general purposes.

3 Defining (Local) Differential Privacy

Suppose each user holds a document $D = \langle x_i \rangle_{i=1}^{L}$ of L tokens (which can be a character, a subword, a word, or an n-gram), where x_i is from a vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$. For privacy, each user derives a sanitized version $\hat{D}$ by running a common text sanitization mechanism $\mathcal{M}$ over D on local devices. Specifically, $\mathcal{M}$ works by replacing every token x_i in D with a substitution $y_i \in \mathcal{V}$, assuming that x_i itself is unnecessary for NLP tasks while its semantics should be preserved for high utility. The output $\hat{D}$ is then shared with an NLP service provider.

We consider a typical threat model in which each user does not trust any other party and views them as an attacker with access to $\hat{D}$ in conjunction with any auxiliary information (including $\mathcal{M}$).

3.1 (Variants of) Local Differential Privacy

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output spaces. A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is a probabilistic function that assigns a random output $y \in \mathcal{Y}$ to an input $x \in \mathcal{X}$. Every y induces a probability distribution on the underlying space. For sanitizing text, we set both $\mathcal{X}$ and $\mathcal{Y}$ as the vocabulary $\mathcal{V}$.

Definition 1 (ϵ-LDP (Duchi et al., 2013))**.** *Given a privacy parameter $\epsilon \geq 0$, $\mathcal{M}$ satisfies ϵ-local differential privacy (ϵ-LDP) if, for any $x, x', y \in \mathcal{V}$,*

$$\Pr[\mathcal{M}(x) = y] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(x') = y].$$

Given an observed output y, from the attacker's view, the likelihoods y is derived from x and x' are similar. A smaller ϵ means better privacy due to a higher indistinguishability level of output distributions, yet the outputs retain less utility.

ϵ-LDP is a very strong privacy notion for its homogeneous protection over all input pairs. However, this is also detrimental to the utility: no matter how unrelated x and x' are, their output distributions must be similar. As a result, a sanitized token y may not (approximately) capture the semantics of its input x, degrading the downstream tasks.

LDP over metric spaces. To capture semantics, we borrow the relaxed notion of Metric-LDP (MLDP) (Alvim et al., 2018) originally proposed for location privacy (Andrés et al., 2013) with the distance metric $d(\cdot, \cdot)$ between two locations (*e.g.*, Manhattan distance (Chatzikokolakis et al., 2013)).

Definition 2 (MLDP)**.** *Given $\epsilon \geq 0$ and a distance metric $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geq 0}$ over $\mathcal{V}$, $\mathcal{M}$ satisfies*

MLDP or $\epsilon \cdot d(x, x')$-LDP if, for any $x, x', y \in \mathcal{V}$,

$$\Pr[\mathcal{M}(x) = y] \leq e^{\epsilon \cdot d(x,x')} \cdot \Pr[\mathcal{M}(x') = y].$$

When $d(x, x') = 1\ \forall x \neq x'$, MLDP becomes LDP. For MLDP, the indistinguishability of output distributions is further scaled by the distance between the respective inputs. Roughly, the effect of ϵ becomes "adaptive." To apply MLDP, one needs to carefully define the metric d (see Section 4.2).

Incorporating ULDP to further improve utility. Utility-optimized LDP (Murakami and Kawamoto, 2019) (ULDP) also relaxes LDP, which was originally proposed for aggregating ordinal responses. It exploits the fact that different inputs have different sensitivity levels to achieve higher utility. By assuming that the input space is split into *sensitive* and *non-sensitive* parts, ULDP achieves a privacy guarantee equivalent to LDP for *sensitive* inputs.

In our context, more formally speaking, let $\mathcal{V}_S \subseteq \mathcal{V}$ be the set of sensitive tokens common to all users, and $\mathcal{V}_N = \mathcal{V} \setminus \mathcal{V}_S$ be the set of remaining tokens. The output space $\mathcal{V}$ is split into the *protected* part $\mathcal{V}_P \subseteq \mathcal{V}$ and the *unprotected* part $\mathcal{V}_U = \mathcal{V} \setminus \mathcal{V}_P$.

The image of $\mathcal{V}_S$ is restricted to $\mathcal{V}_P$, *i.e.*, a sensitive $x \in \mathcal{V}_S$ can only be mapped to a protected $y \in \mathcal{V}_P$. For text, we can set $\mathcal{V}_S = \mathcal{V}_P$ for simplicity. While a non-sensitive $x \in V_N$ can be mapped to $\mathcal{V}_P$, every $y \in \mathcal{V}_U$ must be mapped from $\mathcal{V}_N$, which helps to improve the utility.

3.2 Our New Utility-optimized MLDP Notion

Among many variants of (L)DP notions, we found the above two variants (*i.e.*, ULDP and MLDP) provide useful insight in quantifying semantics and privacy of text data. We thus formulate the new privacy notion of utility-optimized MLDP (UMLDP).

Definition 3 (UMLDP). *Given $\mathcal{V}_S \cup \mathcal{V}_N = \mathcal{V}$, two privacy parameters $\epsilon, \epsilon_0 \geq 0$, and a distance metric $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geq 0}$, $\mathcal{M}$ satisfies $(\mathcal{V}_S, \mathcal{V}_P, \epsilon, \epsilon_0)$-UMLDP, if*

i) for any $x, x' \in \mathcal{V}$ and any $y \in \mathcal{V}_P$, we have

$$\Pr[\mathcal{M}(x) = y] \leq e^{\epsilon d(x,x') + \epsilon_0} \Pr[\mathcal{M}(x') = y];$$

ii) for any $y \in \mathcal{V}_U$, i.e., from an unprotected set $\mathcal{V}_U$ where $\mathcal{V}_U \cap \mathcal{V}_P = \emptyset$, there is an $x \in \mathcal{V}_N$ such that

$$\Pr[\mathcal{M}(x) = y] > 0,$$
$$\Pr[\mathcal{M}(x') = y] = 0\ \forall x' \in \mathcal{V} \setminus \{x\}.$$

Figure 2 summarizes the treatment of UMLDP. It exhibits "invertibility," *i.e.*, $y \in \mathcal{V}_U$ must be "noise-free" and mapped deterministically. Apart from

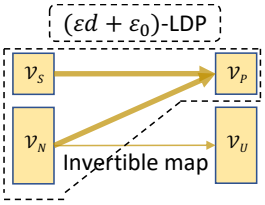

Figure 2: Overview of our new UMLDP notion

generalizing ϵ in the ULDP definition (recalled in Appendix A.1) into $\epsilon d(x, x')$, we incorporate an additive bound ϵ_0 due to the invertibility, which makes the derivation of ϵ easier. Looking ahead, ϵ_0 would appear naturally in the analysis of our UMLDP mechanism for the invertible case.

UMLDP (and MLDP), as an LDP notion, satisfies the *composability* and *free post-processing*. The former means that the sequential execution of ϵ_1-LDP and ϵ_2-LDP mechanisms satisfies $(\epsilon_1 + \epsilon_2)$-LDP, *i.e.*, ϵ can be viewed as the privacy "budget" of a sophisticated task comprising multiple subroutines, each consumes a part of ϵ such that their sum equals ϵ. The latter means further processing the mechanism outputs incurs no extra privacy loss.

4 Our Privacy-Preserving NLP Pipeline

4.1 Overview

We propose two token-wise sanitization methods with (U)MLDP: SANTEXT and SANTEXT$^+$, which build atop a variant of the exponential mechanism (EM) (McSherry and Talwar, 2007) over the "native" text tokens as both input and output spaces to avoid going to the "cursed dimensions" of token embeddings. EM samples a replacement y for an input x based on an exponential distribution, with more "suitable" y's sampled with higher probability (detailed below). It is well-suited for (U)MLDP by considering the "suitability" as how well the semantics of x is preserved for the downstream tasks (run over the sanitized text y) to remain accurate.

To quantify this, we utilize an embedding model mapping tokens into a real-valued vector space. The semantic similarity among tokens can then be measured via the Euclidean distance between their corresponding vectors. Our base design SANTEXT outputs y with probability inverse proportional to the distance between x and y: the shorter the distance, the more semantically similar they are. SANTEXT$^+$ considers some tokens $\mathcal{V}_N$ in $\mathcal{V}$ are non-sensitive, and runs SANTEXT over the sensitive part $\mathcal{V}_S = V \setminus \mathcal{V}_N$ (*i.e.*, it degenerates to SANTEXT if $\mathcal{V}_S = \mathcal{V}$). For $\mathcal{V}_N$, we tailor a probability distribution to provide UMLDP as a whole.

Algorithm 1: Base Mechanism SANTEXT

Input: A private document $D = \langle x_i \rangle_{i=1}^{L}$, and a privacy parameter $\epsilon \geq 0$
Output: Sanitized document $\hat{D}$

Derive token vectors $\phi(x_i)$ for $i \in [1, L]$;
for $i = 1, \ldots, L$ **do**
  Run $\mathcal{M}(x_i)$ to sample a sanitized token y_i with probability defined in Eq. (1);
end
Output sanitized $\hat{D}$ as $\langle y_i \rangle_{i=1}^{L}$;

With SANTEXT or SANTEXT$^+$, each user sanitizes D into $\hat{D}$ and uploads it to the service provider for performing any NLP task built atop a pretrained LM, *e.g.*, BERT. Typically, the task pipeline consists of an embedding layer, an encoder module, and task-specific layers, *e.g.*, for classification.

Without the raw text, the utility can degrade; we thus propose two approaches for improving it. The first one is to pretrain only the encoder on the sanitized public corpus to adapt to the noise. It is optional if pretraining is deemed costly. The second is to fine-tune the full pipeline on $\hat{D}$'s, which updates both the encoder and task layers.

4.2 Base Sanitization Mechanism: SANTEXT

In NLP, a common step is to employ an embedding model[1] mapping semantically similar tokens to close vectors in a Euclidean space. Concretely, an embedding model is an injective mapping $\phi : \mathcal{V} \to \mathbb{R}^m$, for dimensionality m. The distance between any two tokens x and x' can be measured by the Euclidean distance of their embeddings: $d(x, x') = d_{\mathsf{euc}}(\phi(x), \phi(x'))$. As ϕ is injective, d satisfies the axioms of a distance metric.

Algorithm 1 lists the pseudo-code of SANTEXT for sanitizing a private document D at the user side. The first step is to use ϕ to derive token embeddings of each token[2] x in D. Then, for each x, we run $\mathcal{M}(x)$ to sample a sanitized y with probability

$$\Pr[\mathcal{M}(x) = y] = C_x \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\mathsf{euc}}(\phi(x), \phi(y))} \quad (1)$$

where $C_x = (\sum_{y' \in \mathcal{V}} e^{-\frac{1}{2}\epsilon \cdot d_{\mathsf{euc}}(\phi(x), \phi(y'))})^{-1}$.

The smaller $d_{\mathsf{euc}}(\phi(x), \phi(y))$, the more likely y is to replace x. To boost the sanitizing efficiency, we can precompute a $|\mathcal{V}| \times |\mathcal{V}|$ probability matrix, where each entry (i, j) denotes the probability of outputting y_j on input x_i, upon obtaining $\phi(x)$ for $\forall x \in \mathcal{V}$. Lastly, the sanitized $\hat{D} = \langle y_i \rangle_{i=1}^{L}$ can be released to the service provider for NLP tasks.

[1] We assume that it has been trained on a large public corpus and shared by all users.

[2] For easy presentation, we omit the subscript i later.

Algorithm 2: Enhanced SANTEXT$^+$

Input: A private document $D = \langle x_i \rangle_{i=1}^{L}$, a privacy parameter $\epsilon \geq 0$, probability p for a biased coin, and sensitive $\mathcal{V}_S$
Output: Sanitized document $\hat{D}$

Derive token vectors $\phi(x_i)$ for $i \in [1, L]$;
for $i = 1, \ldots, L$ **do**
  if $x_i \in \mathcal{V}_S$ **then**
    Sample a substitution $y_i \in \mathcal{V}_P = \mathcal{V}_S$ with probability given in Eq. (1) ▷ Run SANTEXT over $\mathcal{V}_S$ and $\mathcal{V}_P$;
  else
    Output $y_i = x_i$ with prob. $(1 - p)$; or $y_i \in \mathcal{V}_P$ with prob. in Eq. (2);
  end
end
Output sanitized $\hat{D}$ as $\langle y_i \rangle_{i=1}^{L}$;

4.3 Enhanced Mechanism: SANTEXT$^+$

In SANTEXT, all tokens in $\mathcal{V}$ are treated as sensitive, which leads to excessive protection and utility loss. Following the less-is-more principle, we divide $\mathcal{V}$ into $\mathcal{V}_S$ and $\mathcal{V}_N$, and focus on protecting $\mathcal{V}_S$.

Observing that most frequently used tokens (*e.g.*, a/an/the) are non-sensitive to virtually all users, we use token frequencies for division. A simple strategy, which is also used in our experiments, is to mark the top w of low-frequency tokens (according to a certain corpus) as $\mathcal{V}_S$, where w is a tunable parameter. Looking ahead, this "basic" method already showed promising results. (Further discussion can be found in Section 4.5).

Algorithm 2 lists the pseudo-code of SANTEXT$^+$ with $\mathcal{V}_S = \mathcal{V}_P$ and $\mathcal{V}_N = \mathcal{V}_U$ shared by all users. The first step, as in SANTEXT, is to derive the token embeddings in D. Then, for each token x, if it is in $\mathcal{V}_S$, we sample its substitution y from $\mathcal{V}_P$ with probability given in Eq. (1). (This is equivalent to running SANTEXT over $\mathcal{V}_S$ and $\mathcal{V}_P$.) For $x \in \mathcal{V}_N$, we toss a biased coin. With probability $(1 - p)$, we output y as x (*i.e.*, the "invertibility"). Otherwise, we sample $y \in \mathcal{V}_P$ with probability

$$\Pr[\mathcal{M}(x) = y] = p \cdot C_x \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\mathsf{euc}}(\phi(x), \phi(y))} \quad (2)$$

where $C_x = (\sum_{y' \in \mathcal{V}_P} e^{-\frac{1}{2}\epsilon \cdot d_{\mathsf{euc}}(\phi(x), \phi(y'))})^{-1}$.

As in SANTEXT, we can also precompute two $|\mathcal{V}_S| \times |\mathcal{V}_P|$ and $|\mathcal{V}_N| \times |\mathcal{V}_P|$ probability matrices, which correspond to Eq. (1) and (2), for optimizing

the sanitizing efficiency. Lastly, the sanitized $\hat{D}$ of $\langle y \rangle_{i=1}^{L}$ can be released to the service provider.

Theorem 1. *Given $\epsilon \geq 0$ and d_{euc} over the embedding space ϕ of $\mathcal{V}$,* SANTEXT *satisfies MLDP.*

Theorem 2. *Given $(\mathcal{V}_S = \mathcal{V}_P) \subseteq \mathcal{V}$, $\epsilon \geq 0$, $\epsilon_0 = \ln \frac{1}{p} \geq 0$, and d_{euc} over the embedding space ϕ of $\mathcal{V}$,* SANTEXT^{+} *satisfies $(\mathcal{V}_S, \mathcal{V}_P, \epsilon, \epsilon_0)$-UMLDP.*

Their proofs are in Appendix A.2.

4.4 NLP over Sanitized Text

With $\hat{D}$'s (shared by the users), the service provider can perform any NLP task. In this work, we focus on those built on a pretrained LM, and in particular, we study BERT as an example due to its wide adoption and superior performance. The full NLP pipeline is deployed at the service provider.

Given a piece of (sanitized) text, the embedding layer maps it to a sequence of token embeddings. The encoder computes a sequence representation from the token embeddings, allowing task-specific layers to make predictions. For example, the task layer could be a feed-forward neural network for multi-label classification of a diagnosis system.

The injected noise deteriorates the performance of downstream tasks as the service provider cannot access the raw texts $\{D\}$. To mitigate this, we propose two approaches – pretraining the encoder and fine-tuning the full pipeline, which allow the tasks to be "adaptive" to the noise to some extent.

Pretraining BERT over sanitized public corpus. Besides $\hat{D}$'s, the service provider can also obtain a massive amount of text that is publicly available (say, the English Wikipedia). It also has access to the sanitization mechanisms, and it can produce the sanitized public text (as how users produce $\hat{D}$'s).

Our key idea is to let the service provider pretrain the encoder (*i.e.*, BERT) over the sanitized public text, making it more "robust" in handling $\hat{D}$'s. We thus initialize the encoder with the original BERT checkpoint and conduct further pretraining with an adapted masked language model (MLM) loss. In more detail, the adapted MLM objective is to predict the *original masked tokens* given the sanitized context instead of the one from the raw public text. We note that this is beneficial for improving the task utility, yet may breach the user privacy as the objective learns to "recover" the original tokens or semantics. In Section 5.4, our results will show that such pretrained BERT indeed improves accuracy, with comparable privacy as in original BERT.

Fine-tuning the full NLP pipeline. After pretraining BERT using sanitized public text, the service provider can further improve the efficacy of downstream tasks by fine-tuning the full pipeline. We assume that the ground-truth labels are available to the service provider, say, inferring from $\hat{D}$'s when they can preserve similar semantics to the raw text. Then, the sanitized text-label pairs are used for training/fine-tuning downstream task models, with gradients back-propagated to update the parameters of both the encoder and task layer. We leave more realistic/complex labeling processes based on sanitized texts as future work.

4.5 Definition of "Sensitivity"

Simply treating the top w of least frequent tokens (*e.g.*, according to a public reference corpus) as the sensitive token set already led to promising results (see Section 5.2). By this definition, stop words are mostly non-sensitive (*e.g.*, for $w = 0.9$ over the sentiment classification dataset we used, $\sim$98% of the stop words are deemed non-sensitive). For context-specific corpus, this strategy is better than merely using stop words, *e.g.*, breast cancer becomes non-sensitive among breast-cancer patients.

Sophisticated machine-learning approaches or other heuristics could also be considered, *e.g.*, training over context-specific reference corpus or identifying tokens with personal (and hence sensitive) information (*e.g.*, names). We leave as future work.

Moreover, the definition of sensitivity may vary across users. Some may consider a token deemed non-sensitive by most other users sensitive. The original ULDP work (Murakami and Kawamoto, 2019) has discussed a personalized mechanism that preprocesses such tokens by mapping them to a set of semantic tags, which are the same for all users. These tags will be treated as sensitive tokens for the ULDP mechanism. Apparently, this approach is application-specific and may not be needed in some applications; hence we omit it in this work.

5 Experiments

5.1 Experimental Setup

We consider three representative downstream NLP tasks (datasets) with privacy implications.

Sentiment Classification (SST-2). When people write online reviews, especially the negative ones, they may worry about having their identity traced via writing too much that may provide hints of authorship or linkage to other online writings. For

Mechanisms	SST-2			MedSTS			QNLI		
	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 3$
Random	0.4986	0.4986	0.4986	0.0196	0.0196	0.0196	0.5152	0.5152	0.5152
Feyisetan et al. (2020)	0.5099	0.5143	0.5345	0.0201	0.0361	0.0452	0.5162	0.5256	0.5333
SANTEXT	0.5101	0.5838	0.8374	0.0351	0.5392	0.8159	0.5372	0.5598	0.8116
SANTEXT$^+$	**0.7796**	**0.7943**	**0.8516**	**0.4965**	**0.7082**	**0.8162**	**0.7699**	**0.7760**	**0.8131**
Unsanitized	0.9251			0.8527			0.9090		

Table 1: Utilities comparison of sanitization mechanisms under similar privacy levels using the GloVe embedding

this task, we use the preprocessed version in GLUE benchmark (Wang et al., 2019) of (binary) Stanford Sentiment Treebank (SST-2) dataset (Socher et al., 2013). *Accuracy* (w.r.t. the ground truth included in the dataset) is used as the evaluation metric.

Medical Semantic Textual Similarity (MedSTS). Automated processing of patient records is a significant research direction, and one such task is computing the semantic similarity between clinical text snippets for the benefit of reducing the cognitive burden. We choose a very recent MedSTS dataset (Wang et al., 2020) for this task, which assigns a numerical score to each pair of sentences, indicating the degree of similarity. We report the *Pearson correlation coefficient* (between predicted similarities and human judgments) for this task.

Question Natural Language Inference (QNLI). Question-answering (QA) aims to automatically answer user questions based on documents. We consider a simplified setting of QA, namely QNLI, which predicts whether a given document contains the answer to the question. We use the QNLI dataset from GLUE benchmark (Wang et al., 2019).

We implement our sanitized mechanisms using Python and the sanitization-aware training using the Transformers library (Wolf et al., 2020). We use sanitized data to train and test prediction models for all three tasks. We either build vocabularies for the tasks using GloVe embeddings (Pennington et al., 2014) or adopt the same BERT vocabulary (Devlin et al., 2019). Table 2 shows their sizes. Our sanitization-aware pretraining uses WikiCorpus (English version, a 2006 dump, 600M words) (Reese et al., 2010). We start from the `bert-base-uncased` (instead of randomly initialized) model to accelerate the pretraining.

We set the maximum sequence length to 512, training epoch to 1, batch size to 6, learning rate to 5e-5, warmup steps to 2000, and MLM probability to 0.15. Our sanitization-aware fine-tuning uses the `bert-base-uncased` model for SST-2/QNLI, and `ClinicalBERT` (Alsentzer et al., 2019) for MedSTS. We set the maximum sequence length to 128, training epochs to 3, batch size to 64 for SST-2/QNLI or 8 for MedSTS, and learning rate to 2e-5 for SST-2/QNLI or 5e-5 for MedSTS. Other hyperparameters are kept default. Our hyperparameters followed the transformer library (Wolf et al., 2020) and popular setups in the original dataset literature (Wang et al., 2019, 2020).

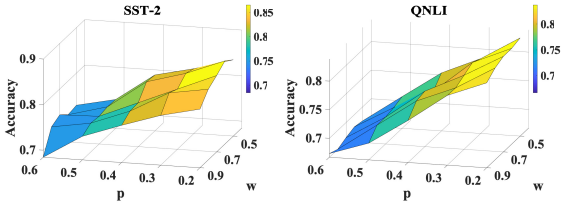

Figure 3: Performance of SANTEXT$^+$ over (w, p) when fixing $\epsilon = 2$ based on the GloVe embedding

5.2 Comparison of Sanitization Mechanisms

We first compare our SANTEXT and SANTEXT$^+$ with random sanitization and the state-of-the-art of Feyisetan et al. (FBDD). Here, we use the GloVe embedding as in FBDD for a fair comparison. Random sanitization picks a token from the vocabulary uniformly. We set the UMLDP parameters $p = 0.3, w = 0.9$ for SANTEXT$^+$ (while Figure 3 plots the impacts of p and w when fixing $\epsilon = 2$).

Table 1 shows the utility of the four mechanisms for the three selected tasks at different privacy levels. FBDD has a higher utility than random replacements. While both FBDD and SANTEXT are based on word embeddings, SANTEXT does not suffer from the "curse-of-dimensionality" and achieves better utility at the same privacy level. SANTEXT$^+$ achieves the best utilities in all cases since it allows the non-sensitive tokens to be noise-free, lowering the noise and improving the utility.

In terms of efficiency, our SANTEXT and SANTEXT$^+$ are very efficient (*e.g.*, ∼2min for the SST-2 dataset) compared with FBDD (∼117min) when they all run on a 24 core CPU machine. This is because our mechanisms only need to compute the sampling probability once and use the same probability matrix for sampling each time, while FBDD needs to recalculate the additive noise and re-search the nearest neighbor each time.

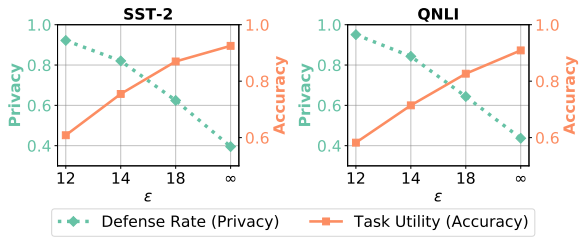

Figure 4: Privacy and Utility Tradeoffs of SANTEXT in terms of Defense Rate (of the Mask Token Inference Attack) versus Accuracy ($\epsilon = \infty$ means "unsanitized.")

Datasets	GloVe embeddings		BERT embeddings	
	$\mathcal{V}$	$\mathcal{V}_S$	$\mathcal{V}$	$\mathcal{V}_S$
SST-2	14,730	13,258	30,522	27,469
MedSTS	3,320	2,989		
QNLI	88,159	79,343		

Table 2: Sizes of vocabularies ($w = 0.9$ for $\mathcal{V}_S$)

5.3 Mask Token Inference Attack

From now on, we adopt the BERT embedding for its superiority. As (U)MLDP is distance-metric dependent, we need to use different ϵ's (*e.g.*, Figure 5) to ensure a similar privacy level, specifically, $\epsilon \cdot d$.

Our sanitization mechanisms provide broad protection for seen/unseen attacks at a fundamental level (by sampling noise to directly replace original tokens) with formally-proven DP, *e.g.*, two guesses of the original token with different styles are nearly probable in an attempt of authorship attribution (Weggenmann and Kerschbaum, 2018) or other "indirect" attacks. Here, we consider a *mask token inference attack* as a representative study to "confirm the theory" by empirically measuring the "concrete" privacy level of sanitized texts.

To infer or recover original tokens given the sanitized text, one can let a pretrained BERT model infer the `MASK` token given its contexts. After all, BERT models are trained via masked language modeling. For each sanitized text of the downstream (private) corpus, we replace each token sequentially by the special token `[MASK]` and input the masked text to the pretrained BERT model to obtain the prediction of the `[MASK]` position. Then, we compare the predicted token to the original token in the raw text. Figure 4 reports the defense rate (the proportion of unmatched tokens to total tokens) and task utility of sanitized texts (by SANTEXT) as well as unsanitized texts on SST-2 and QNLI. We see a privacy-utility trade-off: the more restrictive the privacy guarantee (smaller ϵ), the lower the utility score. Notably, we improve the defense rate substantially with only a small amount of privacy loss (*e.g.*, when $\epsilon = 16$, SANTEXT im-

Datasets	ϵ	Utility		Δ_{privacy}
		Original	**+Pretrain**	
SST-2	12	0.6084	**0.6208**	0.0089
	14	0.7548	**0.7731**	0.0101
	16	0.8698	**0.8830**	-0.0046
QNLI	12	0.5822	**0.6037**	0.0076
	14	0.7143	**0.7309**	-0.0047
	16	0.8265	**0.8369**	-0.0039

Table 3: Sanitization-aware pretraining via SANTEXT proves the defense rate by 20% with only 4% task utility loss over the SST-2 dataset in Figure 4).

5.4 Effectiveness of Pretraining

We then show how the sanitization-aware pretraining further improves the utility but does not hurt the original privacy. Specifically, Table 3 compares the accuracy of sanitization-aware fine-tuning based on the publicly-available `bert-base-uncased` model and our sanitization-aware pretrained one at different privacy levels on SST-2 and QNLI. Our sanitization-aware pretrained BERT models can obtain a 2% absolute gain on average. We conjecture that it can be improved since our pretraining only uses 1/6 of the data used in the original BERT pretraining and 1 training epoch as an illustration.

To demonstrate that such utility improvement is not obtained by sacrificing privacy, we record the change of defense rate (Δ_{privacy}) in launching mask token inference attacks on the original BERT models and our sanitization-aware pretrained BERT models. As Table 3 confirmed, the privacy level of our sanitization-aware pretrained model is nearly the same as the original (sometimes even better).

5.5 Influence of Privacy Parameter ϵ

We aim at striking a nice balance between privacy and utility by tuning ϵ. To empirically show the influence of ϵ, we report the utility and privacy scores over the SST-2 dataset based on SANTEXT. The utility score is the accuracy over the test set. We define three metrics to "quantify" privacy. Firstly, $N_x = \Pr[\mathcal{M}(x) = x]$, which we estimate by the frequency of seeing no replacement by $\mathcal{M}()$. The output distribution of x has full support over $\mathcal{V}$, *i.e.*, $\Pr[\mathcal{M}(x) = y] > 0$ for any $y \in \mathcal{V}$. Yet, we are interested in the effective support $\mathcal{S}$, a set of y's with cumulative probability larger than a threshold, and then define S_x as its size. S_x can be estimated by the number of distinct tokens mapped from x. Both N_x and S_x can be related to two extremes of the Rényi entropy (Rényi, 1961), defined as

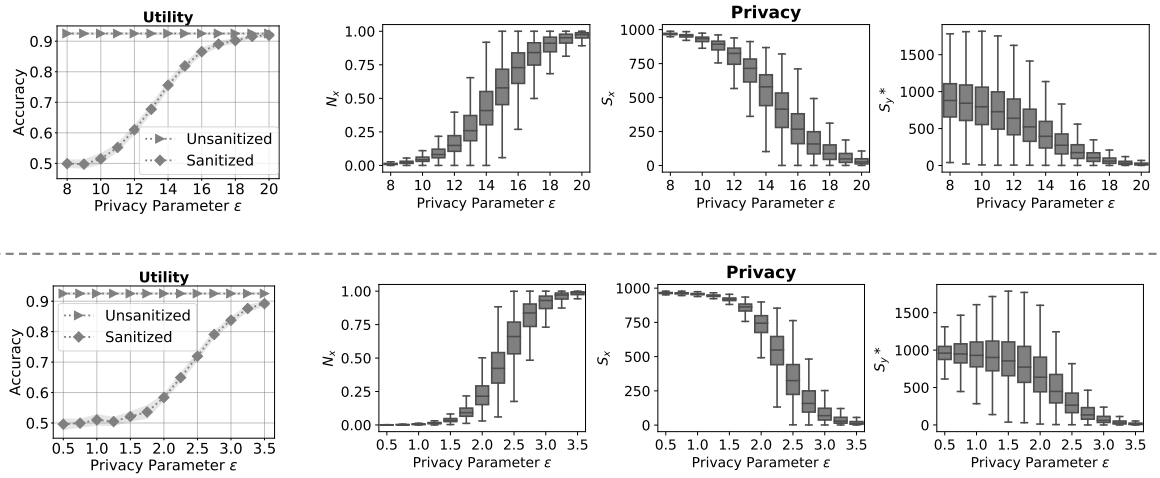

Figure 5: Influence of privacy parameter ϵ of SANTEXT on the utility and privacy (N_x, S_x, S_y^*) based on the SST-2 dataset: The top panel is based on BERT embeddings, and the bottom panel is based on GloVe embeddings.

$H_\alpha(\mathcal{M}(x)) = \frac{1}{1-\alpha}\log(\sum_{y\in\mathcal{V}}\Pr[\mathcal{M}(x)=y]^\alpha)$, with an order $\alpha \geq 0$ and $\alpha \neq 1$. The two extremes are obtained by setting $\alpha = 0$ and $= \infty$, resulting in the Hartley entropy H_0 and the min-entropy H_∞. This implies that we can also approximate H_0 and H_∞ by $\log S_x$ and $-\log N_x$, respectively. Making them large increases the entropy of the distribution.

Another important notion is *plausible deniability* (Bindschaedler et al., 2017), *i.e.*, a set of x's could have led to an output y with a similar probability. We define S_y^* as the set size, estimated by the number of distinct tokens mapped to y.

We run SANTEXT $1,000$ times for the whole SST-2 dataset vocabulary. As Figure 5 shows, when ϵ increases, the utility boosts and N_x increases while S_x, S_y^*, and the privacy level of the mechanism decrease, which gives some intuition in picking ϵ, *e.g.*, for $\sim$40% probability of replacing each token to a different one based on the BERT embeddings (top panel), we could set $\epsilon = 15$ since the median of N_x is $\sim$60% and the accuracy is $\sim$81%.

6 Conclusion

Great predictive power comes with great privacy risks. The success of language models enables inference attacks. There are only a few works in differentially private (DP) text sanitization, probably due to its intrinsic difficulty. A new approach addressing the (inherent) limitation (*e.g.*, in generality) of existing works is thus needed.

Theoretically, we formulate a new LDP notion, UMLDP, which considers both sensitivity and similarity. While it is motivated by text analytics, it remains interesting in its own right. UMLDP enables our natural sanitization mechanisms without the curse of dimensionality faced by existing works.

Practically, we consider the whole PPNLP pipeline and build in privacy at the root with our sanitization-aware pretraining and fine-tuning. With our simple and clear definition of sensitivity, our work already achieved promising performance. Future research in sophisticated sensitivity measures will further strengthen our approach.

Surprisingly, our PPNLP solution is discerning like a cryptographic solution: it is kind (maintains high utility) to the good but not as helpful to the bad (not boosting up inference attacks). We hope our results with different metrics for quantifying privacy can provide more insights in privacy-preserving NLP and make it accessible to a broad audience.

Acknowledgements

Authors at OSU are sponsored in part by the PCORI Funding ME-2017C1-6413, the Army Research Office under cooperative agreements W911NF-17-1-0412, NSF Grant IIS1815674, NSF CAREER #1942980, and Ohio Supercomputer Center (OSC, 1987). The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

Sherman Chow's research is supported by General Research Fund (CUHK 14210319) of UGC, HK. Authors at CUHK would like to thank Florian Kerschbaum for his inspiring talk given at CUHK that stimulated this research.

References

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical BERT embeddings. In *2nd Clinical Natural Language Processing Workshop*, pages 72–78. Also available at arXiv:1904.03323.

Mário S. Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Anna Pazii. 2018. Local differential privacy on metric spaces: Optimizing the trade-off with utility. In *CSF*, pages 262–267.

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

Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. 2017. Plausible deniability for privacy-preserving data synthesis. *PVLDB*, 10(5):481–492.

Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, and Florian Tramèr. 2020a. An attack on InstaHide: Is private learning possible with instance encoding? arXiv:2011.05315.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020b. Extracting training data from large language models. arXiv:2012.07805.

Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the scope of differential privacy using metrics. In *PETS*, pages 82–102.

Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. 2018. Privacy-preserving neural representations of text. In *EMNLP*, pages 1–10.

Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. 2018. Marginal release under local differential privacy. In *SIGMOD*, pages 131–146.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2013. Local privacy and statistical minimax rates. In *FOCS*, pages 429–438.

Cynthia Dwork. 2006. Differential privacy. In *ICALP*, pages 1–12.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *EMNLP*, pages 11–21.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *WSDM*, pages 178–186.

Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. 2019. Leveraging hierarchical representations for preserving privacy and utility in text. In *ICDM*, pages 210–219.

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, and Sanjeev Arora. 2020. TextHide: Tackling data privacy for language understanding tasks. In *EMNLP (Findings)*, pages 1368–1382.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. In *ACL*, pages 25–30.

Qian Lou, Bo Feng, Geoffrey Charles Fox, and Lei Jiang. 2020. Glyph: Fast and accurately training deep neural networks on encrypted data. In *NeurIPS*.

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

Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. 2020b. Towards differentially private text representations. In *SIGIR*, pages 1813–1816.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *ICLR (Poster)*.

Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *FOCS*, pages 94–103.

Takao Murakami and Yusuke Kawamoto. 2019. Utility-optimized local differential privacy mechanisms for distribution estimation. In *USENIX Security*, pages 1877–1894.

Lucien K. L. Ng and Sherman S. M. Chow. 2021. GForce: GPU-friendly oblivious and rapid neural network inference. In *USENIX Security*.

Lucien K. L. Ng, Sherman S. M. Chow, Anna P. Y. Woo, Donald P. H. Wong, and Yongjun Zhao. 2021. Goten: GPU-Outsourcing Trusted Execution of Neural Network Training. In *AAAI*.

OSC. 1987. Ohio supercomputer center.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy risks of general-purpose language models. In *S&P*, pages 1314–1331.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543. 300-d GloVe embeddings available at `http://nlp.stanford.edu/data/glove.840B.300d.zip`.

Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Privacy-adaptive BERT for natural language understanding. arXiv:2104.07504.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI report*.

Samuel Reese, Gemma Boleda, Montse Cuadros, Lluís Padró, and German Rigau. 2010. Wikicorpus: A word-sense disambiguated multilingual wikipedia corpus. In *LREC*.

Alfréd Rényi. 1961. On measures of entropy and information. In *Berkeley Symposium on Mathematical Statistics and Probability*, pages 547–561.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *S&P*, pages 3–18.

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

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *CCS*, pages 377–390.

Latanya Sweeney. 2015. Only you, your doctor, and many others may know. *Technology Science*, 2015092903(9):29.

Florian Tramèr and Dan Boneh. 2019. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *ICLR*.

Sameer Wagh, Shruti Tople, Fabrice Benhamouda, Eyal Kushilevitz, Prateek Mittal, and Tal Rabin. 2021. FALCON: Honest-majority maliciously secure framework for private deep learning. *PoPETs*, 2021(1):187–207.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR (Poster)*.

Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally differentially private protocols for frequency estimation. In *USENIX Security*, pages 729–745.

Tianhao Wang, Ninghui Li, and Somesh Jha. 2018. Locally differentially private frequent itemset mining. In *S&P*, pages 127–143.

Yanshan Wang, Naveed Afzal, Sunyang Fu, Liwei Wang, Feichen Shen, Majid Rastegar-Mojarad, and Hongfang Liu. 2020. MedSTS: a resource for clinical semantic textual similarity. *Lang. Resour. Evaluation*, 54(1):57–72.

Benjamin Weggenmann and Florian Kerschbaum. 2018. SynTF: Synthetic and differentially private term frequency vectors for privacy-preserving text mining. In *SIGIR*, pages 305–314.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP (System Demonstrations)*, pages 38–45. Also available at arXiv:1910.03771, and https://github.com/huggingface.

Harry W. H. Wong, Jack P. K. Ma, Donald P. H. Wong, Lucien K. L. Ng, and Sherman S. M. Chow. 2020. Learning model with error - exposing the hidden model of BAYHENN. In *IJCAI*, pages 3529–3535.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard H. Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *NeurIPS*, pages 585–596.

A Supplementary Formalism Details

A.1 Definition of ULDP

Definition 4 (($\mathcal{V}_S, \mathcal{V}_P, \epsilon$)-ULDP (Murakami and Kawamoto, 2019)). *Given $(\mathcal{V}_S = \mathcal{V}_P) \subseteq \mathcal{V}$, a privacy parameter $\epsilon \geq 0$, $\mathcal{M}$ satisfies $(\mathcal{V}_S, \mathcal{V}_P, \epsilon)$-ULDP if it satisfies the properties:*
i) for any $x, x' \in \mathcal{V}$ and any $y \in \mathcal{V}_P$, we have

$$\Pr[\mathcal{M}(x) = y] \leq e^{\epsilon} \Pr[\mathcal{M}(x') = y];$$

ii) for any $y \in \mathcal{V}_U$, there is an $x \in \mathcal{V}_N$ such that

$$\Pr[\mathcal{M}(x) = y] > 0; \Pr[\mathcal{M}(x') = y] = 0 \text{ for } x \neq x'.$$

A.2 Differential Privacy Guarantee

Proof of Theorem 1. Consider $L = 1$, *i.e.*, $D = \langle x \rangle$. For another document D' with $x' \in \mathcal{V} \setminus \{x\}$ and a possible output $y \in \mathcal{V}$:

$$\begin{aligned}
&\frac{\Pr[\mathcal{M}(x) = y]}{\Pr[\mathcal{M}(x') = y]} \\
=&\frac{C_x \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\text{euc}}(\phi(x), \phi(y))}}{C_{x'} \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\text{euc}}(\phi(x'), \phi(y))}} \\
=&\frac{C_x}{C_{x'}} \cdot e^{\frac{1}{2}\epsilon \cdot [d(x', y) - d(x, y)]} \\
\leq&\frac{C_x}{C_{x'}} \cdot e^{\frac{1}{2}\epsilon \cdot d(x, x')} \\
=&\frac{\sum_{y' \in \mathcal{V}} e^{-\frac{1}{2}\epsilon \cdot d(x', y')}}{\sum_{y' \in \mathcal{V}} e^{-\frac{1}{2}\epsilon \cdot d(x, y')}} \cdot e^{\frac{1}{2}\epsilon \cdot d(x, x')} \\
\leq&e^{\frac{1}{2}\epsilon \cdot d(x, x')} \cdot e^{\frac{1}{2}\epsilon \cdot d(x, x')} = e^{\epsilon \cdot d(x, x')}
\end{aligned}$$

The proof, showing SANTEXT ensures $\epsilon \cdot d(x, x')$-LDP, mainly relies on the triangle inequality of d. To generalize to the case of $L > 1$, we sanitize every token x_i in D independently, and thus:

$$\Pr[\mathcal{M}(D) = \hat{D}] = \prod_{i=1}^{L} \Pr[\mathcal{M}(x_i) = y_i].$$

Then, for any D, D', the privacy bound is given as

$$\frac{\Pr[\mathcal{M}(D) = \hat{D}]}{\Pr[\mathcal{M}(D') = \hat{D}]} \leq e^{\epsilon \cdot \sum_{i=1}^{L} d(x_i, x'_i)},$$

which follows from the composability. □

Proof of Theorem 2. Consider the case $L = 1$ with $D = x$ and $D' = x'$. For $x, x' \in \mathcal{V}_S$, the output y is restricted to $\mathcal{V}_P$, with the proof identical to the above theorem (as SANTEXT is run over $\mathcal{V}_S, \mathcal{V}_P$).

For $x, x' \in \mathcal{V}_N$ and $y \in \mathcal{V}_P$, we have

$$\begin{aligned}
\frac{\Pr[\mathcal{M}(x) = y]}{\Pr[\mathcal{M}(x') = y]} &= \frac{p \cdot C_x \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\text{euc}}(\phi(x), \phi(y))}}{p \cdot C_{x'} \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\text{euc}}(\phi(x'), \phi(y))}} \\
&\leq e^{\epsilon \cdot d(x, x')}.
\end{aligned}$$

For $x \in \mathcal{V}_S$, $x' \in \mathcal{V}_N$, and $y \in \mathcal{V}_P$, we have

$$\begin{aligned}
\frac{\Pr[\mathcal{M}(x) = y]}{\Pr[\mathcal{M}(x') = y]} &= \frac{C_x \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\text{euc}}(\phi(x), \phi(y))}}{p \cdot C_{x'} \cdot e^{-\frac{1}{2}\epsilon \cdot d_{\text{euc}}(\phi(x'), \phi(y))}} \\
&\leq \frac{1}{p} \cdot e^{\epsilon \cdot d(x, x')} = e^{\epsilon \cdot d(x, x') + \epsilon_0}.
\end{aligned}$$

The probability for $x \in \mathcal{V}_N$ is $(1-p)$. The above inequalities thus show that SANTEXT$^+$ ensures the properties of UMLDP. Similarly, we use the composability to generalize for $L > 1$. □

A.3 Qualitative Observations

Below, we focus on SANTEXT sanitizing a single token x. We first make two extreme cases explicit. (1) When $\epsilon = 0$, the distribution in Eq. (1) becomes $\Pr[\mathcal{M}(x) = y] = \frac{1}{|\mathcal{V}|}, \forall y \in \mathcal{V}$. SANTEXT is perfectly private since y is uniformly sampled at random, independent of x. Yet, such a y does not preserve any information of x.
(2) When $\epsilon \to \infty$, we have $\Pr[\mathcal{M}(x) = x] \gg \Pr[\mathcal{M}(x) = y], y \in \mathcal{V} \setminus \{x\}$. $\Pr[\mathcal{M}(x) = x]$ dominates others since $d(x, x) = 0$ and $d(x, y) > 0$. This loses no utility as x almost stays unchanged, yet provides no privacy either.

For a general $\epsilon \in (0, \infty)$, the distribution has full support over $\mathcal{V}$, *i.e.*, we have a non-zero probability for any possible $y \in \mathcal{V}$ such that $\mathcal{M}(x) = y$. Also, given $y, y' \in \mathcal{V}$ with $d(x, y) < d(x, y')$, we have $\Pr[\mathcal{M}(x) = y] > \Pr[\mathcal{M}(x) = y']$. As ϵ increases, $\Pr[\mathcal{M}(x) = y]$ for the y's with large $d(x, y)$ goes smaller (and even approaches 0). This means that the output distribution becomes "skewed," *i.e.*, the outputs concentrate on those y's with small $d(x, y)$. This is good for utility, which stems from the semantics preservation of every token. On the contrary, too much concentration weakens the privacy.

For SANTEXT$^+$, the above results directly apply to the case $x \in V_S$ (as SANTEXT is run over $\mathcal{V}_S$ and $\mathcal{V}_P$). There is an extra p determining whether a $x \in \mathcal{V}_N$ is mapped to a $y \in \mathcal{V}_P$. If so, the results are similar except with an extra multiplicative p. A larger p leads to stronger privacy as the probability $(1 - p)$ of x being unchanged becomes smaller.

Dataset: SST-2		
Mechanisms	ϵ	**Original Text:** it 's a charming and often affecting journey .
SANTEXT	1	heated collide. charming activity cause challenges beneath tends
	2	worse beg, charming things working noticed journey basically
	3	all 's. charming and often already journey demonstrating
SANTEXT$^+$	1	it unclear a charming and often hounds journey
	2	it exaggeration a charming feelings often lags journey .
	3	it 's a tiniest picked often affecting journey .
Dataset: QNLI		
Mechanisms	ϵ	**Original Text:** When did Tesla move to New York City? In 1882, Tesla began working for the Continental Edison Company in France, designing and making improvements to electrical equipment.
SANTEXT	1	43 trapper Gaga MCH digest sputtering avenged Forced Laborers Homage Ababa afer psychic 51,000 intercity lambasting nightmare–confederate Frontier Britian Manor Londres shards pilot Mining faster alone Thessalonica Bessemer Lie Columbus
	2	blame least ethos did tenth ballot Condemnation critical filmed In 1883 3200 Conversion pushing 7:57 enabling Town stamp Time downwards Peterson France, GSA emulating addresses appealing 47.4 electrical pull refreshing
	3	Wave did Tesla It way Dru Tully breaking? Tupelo 1875, Tesla began escaped for announcing Continental Edison Company in France However designing and making improvements to electrical Chongqing add
SANTEXT$^+$	1	Rodgers did Sung move to New plantation City ? In K. innumerable Gunz began working sliding the Sultans Edison Company structured France beaching designing disseminate making tribunals to lackluster equipment 40-foot
	2	vaults did Tesla chunks introduces Teknologi Eyes City ? In 866 , Tesla began working for the Analytical Edison Company Butterfly France , designing Sias siblings Noting circumventing electrical orient .
	3	When did Tesla guideline to New York City ? In 1885 , Tesla MG working for the Continental Edison Company in France , translating and dreamed improvements ascertain electrical lookout .

Table 4: Qualitative examples from the SST-2 and QNLI datasets: Sanitized text by our mechanisms at different privacy levels based on GloVe embeddings

B Qualitative Examples

Table 4 shows two examples of sanitized texts output by SANTEXT and SANTEXT$^+$ at different privacy levels from the SST-2 and QNLI datasets.

C Supplementary Related Works

Privacy is a practically relevant topic that also poses research challenges of diverse flavors. Below, we discuss some "less-directly" relevant works, showcasing some latest advances in AI privacy.

Cryptographic Protection of (Text) Analytics. There has been a flurry of results improving privacy-preserving machine-learning frameworks (*e.g.*, (Lou et al., 2020)), which make use of cryptographic tools such as homomorphic encryption and secure multi-party computation (SMC) for general machine/deep learning. These cryptographic designs can be adapted for many NLP tasks in principle. Nevertheless, they will slow down computations by orders of magnitude since cryptographic tools, especially fully homomorphic encryption, are generally more heavyweight than the DP approaches. One might be tempted to replace cryptography with *ad hoc* heuristics. Unfortunately, it is known to be error-prone (*e.g.*, a recently proposed attack (Wong et al., 2020) can recover model parameters during "oblivious" inference).

A recent trend (*e.g.*, (Wagh et al., 2021)) relies on multiple non-colluding servers to perform SMC for secure training. However, SMC needs multiple rounds of communication. It is thus more desirable to have a dedicated connection among the servers.

Albeit with better utility (than DP-based designs), cryptographic approaches mostly consider immunity against membership inference (Shokri et al., 2017) to be out of their protection scope since DP mechanisms could be applied over the training

data before the cryptographic processing.

There is a growing interest in privacy-preserving analytics in the NLP community too. Very recently, TextHide (Huang et al., 2020) devises an "encryption" layer for the hidden representations. Unfortunately, it is shown to be insecure by cryptographers and privacy researchers Carlini et al. (2020a).

Hardware-Aided Approaches. GPU can compute linear operations in a batch much faster than CPU. Nevertheless, we still need a protection mechanism in using GPU, another protection mechanism for the non-linear operations, and their secure integration. In general, utilizing GPU for privacy-preserving machine-learning computations is non-trivial (*e.g.*, see (Ng and Chow, 2021) for an extended discussion).

To exploit the parallelism of GPU while minimizing the use of cryptography, one can resort to a trusted processor (*e.g.*, Intel SGX) for performing non-linear operations within its trusted execution environment (TEE) Note that one still needs to use cryptographic protocols to outsource the linear computation to (untrusted) GPU. Slalom (Tramèr and Boneh, 2019) is such a solution that supports privacy-preserving inference. Training is a more challenging task that was left as an open challenge. Recently, it is solved by Goten (Ng et al., 2021). Notably, both works are from cryptographers but also get recognized by the AI community.

Finally, we remark that the use of TEE is not a must in GPU-enabled solutions. For example, GForce (Ng and Chow, 2021) is one of the pioneering works that proposes GPU-friendly protocols for non-linear layers with other contributions.