Local Differential Privacy for Privacy-Preserving NLP Tasks

Anonymous ACL submission

Abstract

In this paper, we propose a Local Differentially Private Natural Language Processing (LDP-NLP) model that protects the privacy of user input sentences for both training and inference stages while requiring no server security trust. Compared to existing methods, the novel privacy-preserving methodology significantly reduces calibrated noise power and thus improves model accuracy by incorporating (a) an LDP-layer, (b) sub-sampling and up-sampling DP amplification algorithms for training and inference, and (c) DP composition algorithms for noise calibration. This novel LDP-NLP solution guarantees privacy for the entire training/inference data for the first time, whereas current methods can only guarantee privacy for either a single training/inference step. Furthermore, the total privacy cost is reduced to a reasonable range, i.e., less than 10, for the first time with an accuracy loss of only 2-5% compared to the accuracy upper bound produced by the original model without privacy guarantee.

1 Introduction

Natural Language Processing (NLP) based on (deep) neural architectures has given rise to a new generation of applications such as sentiment analysis, question answering, and information retrieval (Sun et al., 2019; Pang et al., 2017; Chen et al., 2017). The majority of these applications may require a significant amount of personal data during the training stage, as well as personal queries sent to the server during the inference stage, raising a number of privacy concerns. Question-answering systems, for example, require personal data for training (fine-tuning) and questions from the user are required again during the inference stage to query answers from the service provider. Many studies, however, have discovered privacy violations in deep learning models due to the input information embedded in latent representations and model parameters (Shokri et al., 2017; Carlini et al., 2019). To protect privacy, there has been an increase in demand for privacy-preserving NLP model (Feyisetan et al., 2020b).

Data anonymization that removes personally identifiable information or protected attributes from data is insufficient as innocuous-looking attributes can be linked to other information sources for reidentification (Pedreshi et al., 2008; Sweeney, 2015). Differential Privacy (DP) (Dwork et al., 2006), on the other hand, randomizes the computation process to stabilize the output in the face of changes to input data, ensuring that the adversary can hardly tell if an individual data item, i.e., a word or a sentence (which is the protection granularity depending on the mechanism designed), is in the dataset or not by looking at the computation output, i.e., latent representation or model. DP has been integrated into the deep learning training stage (Abadi et al., 2016) by appropriately randomizing backpropagation with calibrated noise to limit what could be breached from the *training data* when revealing the model. This is known as Differentially-Private Stochastic Gradient Descent (DP-SGD). As a result, the model parameters can be viewed as a sanitized release, with individual training data obscured but the model still remains functional. However, due to the calibrated noise required for DP, it has been recognized that DP mechanisms invariably significantly reduce the downstream task performance, raising the privacy-utility tradeoff issue (Dwork and Lei, 2009).

In the context of NLP in this paper, we focus on the practical situation in which users[1] are concerned about the privacy of their sensitive data and the server is untrustworthy. Since the privacy of data embedded in the model/gradient can be violated (Shokri et al., 2017; Carlini et al., 2019),

[1] A user is a data owner in this paper, who could be an individual single user or a curator involved in multi-party computation.

Local DP (LDP) is required to protect user's input sentences before sharing them or sharing their computation results with the server. Furthermore, as the downstream task implemented at the server, the user itself cannot perform DP-SGD-based training to protect the privacy of its training data, let alone the privacy of its inference data. To address the aforementioned problems, we propose a novel LDP layer on the user side to randomize the intermediate output for training's and inference's forward computations, respectively. We successfully push the privacy-accuracy tradeoff boundary significantly by carefully designing noise calibration algorithms based on sampling, achieving for the first time state-of-the-art accuracy while lowering the privacy parameter ϵ to less than 10 for the entire training/inference stage. The technical contributions are summarized below.

(I) We develop a novel non-parametric DP layer along with the noise power calibration algorithms that provides LDP for not only the training sentences but also the inference sentences.

(II) For the first time in the LDP-NLP model, we propose novel sentences sub-sampling and up-sampling DP-amplification mechanisms in the training and inference stages, that reduce the privacy cost parameter ϵ to less than 10 across the entire training/inference data set. By contrast, the same data privacy cost ϵ can only be guaranteed in a single training or inference step in the literature. In other words, thousands of additional training/inference steps can be carried out in the proposed method with the same level of privacy as existing methods.

(III) The proposed generic LDP-NLP methodology significantly outperforms state-of-the-art methods on typical LDP-NLP tasks and benchmarks. In comparison to the performance upper bound produced by the version that does not guarantee privacy, the accuracy loss due to privacy preservation is only 2%-5% for an LSTM or a BERT encoder with privacy cost less than 10.

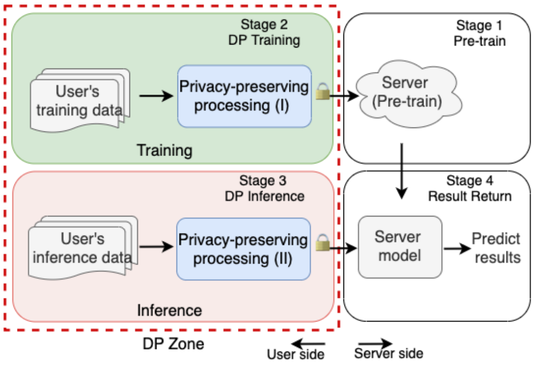
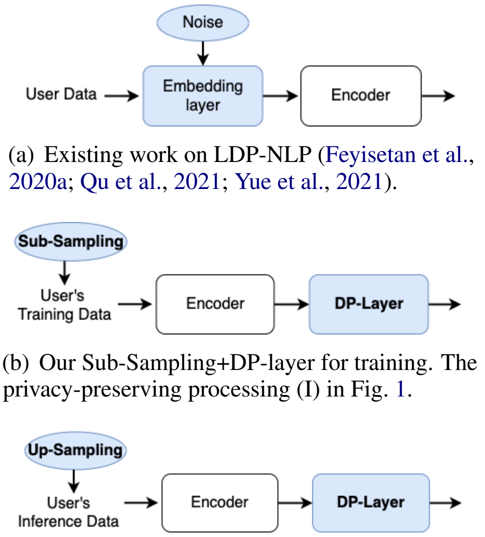


Figure 1: LDP-NLP pipeline for DP training and DP inference in Stages 2 and Stage 3 to protect training and inference.



(a) Existing work on LDP-NLP (Feyisetan et al., 2020a; Qu et al., 2021; Yue et al., 2021).

(b) Our Sub-Sampling+DP-layer for training. The privacy-preserving processing (I) in Fig. 1.

(c) Our Up-Sampling+DP-layer for inference. Privacy-preserving processing (II) in Fig. 1.

Figure 2: Different LDP architectures for the privacy-preserving modules (in blue) in Fig. 1.

2 LDP-NLP Task Pipeline

Fig. 1 depicts a target scenario with corresponding LDP operations in Fig. 2[2]. As the user's input contains sensitive information, our primary requirement is that the service provider only accesses LDP-guaranteed representation, which means that all of the information sent out locally by the user in all steps together must satisfy the DP definition, with little chance of determining whether or not an individual sentence is in the data set. The four stages required to complete an LDP-NLP task between a user and the server are summarized below, with the training (Stage 2) and inference (Stage 3) stages containing user sentence input that requires DP assurance.

Stage 1 (Pre-train): If necessary, the server performs pre-training on its own data.

Stage 2 (DP training): During collaborative training, each user converts raw data (e.g., sentences) into LDP-guaranteed latent representations before sending them to the server. We propose a non-parametric DP layer after the encoder to achieve

[2] Although this study focuses on a single user case, it is easily adaptable to a more general setting in which sensitive data is collected independently from multiple users using LDP.

LDP guarantee with details about the operations and the noise calibration algorithm in Section 4.1 and Section 4.2, respectively.

Stage 3 (DP inference): The raw data is converted by the local DP-layer into DP-guaranteed latent representations before being sent to the server for the inference process. This stage enables some calibrated noise power, which depends on the upsampling DP amplification algorithm. This will be described in Section 4.3.

Stage 4 (Inference result return): The model of the server responds to DP-protected queries.

Note that, after receiving the DP-guaranteed representations in Stage 3 and Stage 4, the server computes the remaining forward computations and backpropagations for both the inference and training stages. Note that DP-SGD in Li et al. (2021); Anil et al. (2021); Dupuy et al. (2021), which is designed to protect only the training data while sharing the training model, is inapplicable in the considered situation because performing the downstream task would require the untrusted server to compute the true gradient, resulting in data leakage (Zhao et al., 2020).

Paper organization This paper focuses primarily on Stage 3 and Stage 4. We will first introduce the DP definition in the context of NLP problems in Section 3, and then in Section 4, we will formulate the DP layer operators as well as the noise calibration algorithms for the training and inference stages. Section 5 contains information about related work. In Section 6, we provide experimental validation.

3 LDP-NLP Model

DP Mechanism: A random DP algorithm's output is stabilized to the point where the presence or absence of any specific data item, such as a word or a sentence in an NLP task, is hardly distinguishable. The type of data item is the *DP granularity*. The greatest possible divergence between the output distributions of DP algorithms when applied to two datasets that differ by arbitrary data item of the DP granularity, i.e., word or sentence, describes the DP protection level (also known as DP cost). More accurately, let x and x' be datasets that differ in one data item (a word or a sentence), then the randomized algorithm $\mathcal{M}$, which is the local user output in our problem (details in Equation (4)), is (ϵ, δ)-DP if for arbitrary subset, i.e., $\mathcal{Y}$ of the all possible output of $\mathcal{M}$:

$$\Pr[\mathcal{M}(x) \in \mathcal{Y}] \leq \mathrm{e}^{\epsilon} \Pr[\mathcal{M}(x') \in \mathcal{Y}] + \delta. \quad (1)$$

This equation shows that the protection level is related to ϵ: the lower ϵ values, the better privacy protection. The value δ can be interpreted as the likelihood of not achieving DP. Specifically, when $\delta = 0$, we get ϵ-DP. We notice that the protection level is also related to the level of granularity: the higher the level of granularity, the higher the level of privacy guaranteed. Therefore the sentence level DP considered in this paper has a better privacy protection than the word level DP proposed in the literature (Feyisetan et al., 2020a; Qu et al., 2021; Yue et al., 2021) in general, because a sequence of words in the sentence is hidden in the former case. However, the specific operation $\mathcal{M}$ in the existing studies cannot be easily extended from word protection to sentence protection. We proposed a DP-layer to achieve the aforementioned goals with details in Section 4.1.

Given a target privacy budget (ϵ, δ) for protecting the entire training and inference stages, the next critical point is to calibrate the noise power, a parameter for that DP mechanism. It should be noted that the lower the noise power, the higher the model's accuracy but the lower the level of privacy. To solve this dilemma, we investigate subsampling/upsampling methods for DP amplification and tight DP composition among training/inference steps in order to reduce noise power without compromising the privacy.

DP Amplification&Composition: Intuitively, privacy amplification by sampling is caused by the fact that an individual sentence has complete privacy if it is not included in the samples and whether or not the sentence is included is a secret. In this paper, we investigate *sub-sampling DP amplification* for each NLP training step and further propose for the first time *up-sampling DP amplification* for the NLP inference/query by introducing fictitious data on the user side.

To complete an NLP task, both the training and query stages must perform a series of computation steps on the private training/inference dataset, with each computation step potentially based on the results of previous computation steps on the same dataset. Even if each step i is DP protected with a privacy cost (ϵ_i, δ_i), taking all steps output together by the adversary may no longer guarantee privacy. The computation of privacy degradation as the number of steps increases is referred to as DP composition. Specifically, since an NLP training stage requires even more than thousands of step updates and the proposed up-sampling inference usually also involves multiple queries, a tight composition methodology is needed. The detailed analysis and corresponding noise calibration algorithms are provided in Section 4.2 and 4.3.

4 Operations in the LDP-NLP Model

4.1 DP Mechanism in the DP-Layer

We propose a nonparametric DP layer by injecting calibrated noise into the output of a clipped encoder. It is important to note that applying the DP layer directly to the embeddings results in significant performance degradation due to the significant loss of semantic information for the downstream task. As a result, an encoder is used before the DP layer on the user side. A light-weight encoder, such as LSTM, can be used for a computation/memory-limited user, while stacked transformers can be used for a powerful user, such as a curator.

Clipping Operation: One method to stablize the output is clipping. Let x and x' be arbitrary pair of sentences from the training or inference sets, and define $f(x)$ as the corresponding output of the encoder in Fig. 2(b) and Fig. 2(c). The sensitivity Δ of $f(x)$, which is the greatest variation output for a pair of sentences with the ℓ_2-norm is given by

$$\Delta = \max_{x,x'} ||f(x) - f(x')||_2. \tag{2}$$

Because of the randomness of training data, computing Δ is difficult. We limit f's output range by clipping per sentence representation from the output of local encoder with

$$\mathrm{CL}\left(f(x);C\right) = f(x)\cdot\min\left(1, \frac{C}{\|f(x)\|_2}\right). \tag{3}$$

The quantity C is a predefined hyper-parameter. The lower the value of C, the less calibrated noise power is required for a given level of DP protection. Cutting too much with a small C, on the other hand, will harm the semantic information embedded and will result in a significant performance drop.

After clipping, we apply additive Gaussian noise to improve the accuracy of the NLP model while still providing DP guarantee. Hence, the DP layer output is given by

$$\mathcal{M}(x, f(\cdot), \epsilon, \delta) = \mathrm{CL}(f(x)) + \mathcal{N}(0, \sigma^2). \tag{4}$$

It is shown that the calibrated noise power σ^2 and the DP profile $(\epsilon(\delta), \delta)$ follows (Dong et al., 2021):

$$\delta(\epsilon;\mu) = \Phi\left(-\frac{\epsilon}{\mu}+\frac{\mu}{2}\right) - \mathrm{e}^{\epsilon}\Phi\left(-\frac{\epsilon}{\mu}-\frac{\mu}{2}\right) \tag{5}$$

with

$$\mu = \Delta/\sigma, \tag{6}$$

where $\Phi(t)$ is the c.d.f. of the standard normal distribution.

Algorithm 1 Non-parametric DP-Layer

```
Require: Latent representation $f(x) \in \mathbb{R}^d$, clipping value $C$, noise variance $\sigma^2$
1: Gaussian Mechanism: $\widetilde{x} \leftarrow \mathrm{CL}\left(f(x);C\right) + z$ with $z \sim \mathcal{N}(0, \sigma^2 I_d)$.
2: return $\widetilde{x}$.
```

In summary, we form a DP-layer containing clipping operation to bound the output sensitivity and additive Gaussian noise on the latent representation. A formal statement for the privacy guarantees of Algorithm 1 is provided in Theorem 1.

Lemma 1. *(DP-Layer Privacy) Let $f(x)$ be the encoder output with ℓ_2 sensitivity C given by Equation (3). For any $\epsilon > 0$ and $\delta \in (0, 1)$, the mechanism described in Algorithm 1 is (ϵ, δ)-DP for each time of using the DP layer.*

We evaluate the privacy cost for each step of input forward though DP-layer in Algorithm 1 based on Gaussian DP (GDP), which measure the privacy profile (ϵ, δ) in terms of μ via (5) and (6). To make the paper self-contained, we introduce the GDP preliminary in the appendix and for more details please refer to Dong et al. (2021). Composition enjoys a simple and convenient formulation in GDP, i.e., the n-fold composition of u_i-GDP mechanisms is $G_{\mu_1} \otimes G_{\mu_2} \otimes \cdots \otimes G_{\mu_n} = G_{\mu}$-DP with $\mu = \sqrt{\mu_1^2 + \cdots + \mu_n^2}$. Let x^t denote the sampled sentences for the t-th update step (training or inference) with $|x^t|$ the number of sentences and x_k^t the k-th sentence. The output of x_k^t after the DP layer is C/σ_t-GDP according to (6). By calibrating the dynamic noise power σ_t for each step, we have the μ_{train}-GDP composition result of all the sampled sentences in each step with

$$\mu_{\text{train}} = \left|x^t\right| \frac{C}{\sigma_t}. \tag{7}$$

In the following, we show how to calibrate the noise by leveraging sub-sampling and up-sampling to conduct DP amplification in the training and inference stages, respectively.

Algorithm 2 Noise calibration (Training stage)

```
Require: Training DP budget $(\epsilon, \delta)$, sampling rate $p_{\text{train}}$, and samples $x_t$ for all $t \in T$.
1: Compute $\mu_{\text{tot}}$ corresponding to $(\epsilon, \delta)$ by (5)
2: Compute $\mu_1 = \cdots = \mu_T = \mu_{\text{train}}$ by (8)
3: Compute $\sigma_t = |x^t| \frac{C}{\mu_{\text{train}}}$ by (6) for all $t \in [T]$.
```

Algorithm 3 Noise calibration (Inference stage)

```
Require: Inference/query DP budget $(\epsilon, \delta)$, true data rate $q$, sampling rate $p_{\text{query}}$
1: Mix the true query data with fictitious data by keeping the true data rate $q$
2: Sample query data sets $x_t$ for all $t \in T$ from the mixed dataset.
3: Compute $\mu_{\text{tot}}$ corresponding to $(\epsilon, \delta)$ by (5)
4: Compute $\mu_1 = \cdots = \mu_T = \mu_{\text{query}}$ by (9)
5: Compute $\sigma_t = |x^t| \frac{C}{\mu_{\text{train}}}$ by (7) for all $t \in [T]$
```

4.2 Noise Calibration for Training

Each update step is performed on a sub-sampled sentences, which is obtained through an independent Bernoulli trial of all sentences with probability p_{train}. The dual function of (5) for each subsamples with DP amplification can be expressed by $p_{\text{train}} \cdot G_{\mu_t} + (1 - p_{\text{train}})\,\text{Id}$ (Dong et al., 2021), with μ_t computed by (6) and $\text{Id} : [0,1] \to [0,1]$ being $\text{Id}\,(\alpha) = 1 - \alpha$. Usually the sub-sampling rate p_{train} is much smaller than 1, and thus the trade-off function is much smaller than G_{μ_t}. Since it does not satisfy GDP, we cannot directly apply n-fold composition of GDP anymore. Consider a series of T adaptive compositions of $p_{\text{train}} \cdot G_{\mu_t} + (1 - p_{\text{train}})\,\text{Id}$. According to the recent central limit theorem for GDP (Bu et al., 2020), the trade-off function for $\lim_{T\to\infty} \otimes_{t=1}^{T} (p_{\text{train}} \cdot G_{\mu_t} + (1 - p_{\text{train}})\,\text{Id}\,)$ approaches $G_{\mu_{\text{tot}}}$-DP, which is given by

$$\mu_{\text{tot}}^{\text{train}} = p_{\text{train}} \cdot \sqrt{T\left(\mathrm{e}^{\mu_{\text{train}}^2} - 1\right)}. \quad (8)$$

In summary, given a privacy budget for the training stage, i.e., (ϵ, δ) and training steps T, we can first subsample the training data sets to construct the mini-batch for each step update and then calibrate the noise power for each step as shown in Algorithm 2.

4.3 Noise Calibration for Inference

To improve utility, we propose DP amplification via upsampling for the inference stage. The general idea is to introduce uncertainty into the inference data set by upsampling it with fictitious data. We generate some fictitious sentences that do not contain any private information and mix them with the true queries before randomly sampling the queries multi-steps to send to the server via the DP layer. Note that all the true queries will be send out via multi-upsamplings. Because of the effectiveness of this type of DP amplification, noise power will be significantly reduced, improving inference accuracy without jeopardizing privacy.

Let x and y denote the original and fictitious inference sets, respectively. Then we have the true sentence rate $q \triangleq |x|/(|x|+|y|)$. For each step, we sample each query sentence in the mixed data set by independent Bernoulli trial with probability p_{query}. Then the probability that each true query is sampled is given by $q \cdot p_{\text{query}}$. Following similar analysis in the previous training stage DP amplification, the GDP privacy parameter for the inference stage is

$$\mu_{\text{tot}}^{\text{query}} = q \cdot p_{\text{query}} \cdot \sqrt{T\left(\mathrm{e}^{\mu_{\text{query}}^2} - 1\right)}. \quad (9)$$

Similar to Algorithm 2, given a privacy budget for the query stage, i.e., (ϵ, δ) and query number T, the noise power calibration for the inference stage is shown in Algorithm 3.

Note that the DP amplification in the inference stage does not come for free. Similarly to how downsampling reduces the training convergence rate, up-sampling increases the query/inference times since the mixed fictitious sentences and uncertainty from the sampling. In the next section, we will put this to the test through experimentation.

5 Related Work

To share the model while protect the corresponding training data privacy, the previous study used DP-SGD to train privacy-preserving models (Shokri and Shmatikov, 2015; Yu et al., 2019). Because of the large number of steps for training, even though each step provides a reasonable DP cost, say, $\epsilon = 3$, the overall privacy cost explodes, which does not provide any privacy guarantee. Abadi et al. (2016) propose the first work with a reasonable level of ϵ for DP-SGD. Because of their moment accounting technique for tight composition, the calibrated noise is much smaller than all previous methods based on the same privacy budget. More recently, Li et al. (2021) study the DP-SGD for NLP problem to reduce the performance loss due to privacy preserving, and Anil et al. (2021); Dupuy et al. (2021)

show how to efficiently train an NLP model via DP-SGD. However, as SGD-based training takes place on the server and is unrelated to inference data in user-server settings, DP-SGD based methods are inapplicable to the LDP-NLP model investigated in this paper.

Another line of research focuses on user data protection in the user-server model, which is also the subject of this paper. Feyisetan et al. (2020a); Qu et al. (2021); Yue et al. (2021) aim to protect the local input words based on the metric DP (Chatzikokolakis et al., 2013), a relaxation of DP definition. The methods can only be applied to the token representation layer by evaluating semantic distance between words in the latent representation space. Due to the fact that these works (Feyisetan et al., 2020a; Qu et al., 2021; Yue et al., 2021) adhere to the metric DP and its corresponding mechanism, tight DP composition/amplification for the metric DP related mechanism remains lacking. Furthermore, sampling words from a data set of sentences using a sampling distribution, such as Poisson sampling or uniform sampling, is difficult. Due to the aforementioned reasons, DP is only guaranteed for each training/inference step, but the DP cost for the entire data set scales to the number of training steps (typically more than thousands steps), which no longer guarantees privacy. Furthermore, the existing methods only provide DP protection at the word level, falling short of a more stringent DP protection requirement, such as sentence level DP protection. We compare our DP-NLP model to existing works in Fig. 2 and Table 1.

6 Experiments

We conduct empirical privacy-utility test on text-matching and classification tasks. Note that the comparison is not fair for the proposed DP-NLP model because we protect an entire sentence rather than just a word as is done in the literature. Despite this, the experimental results show that the proposed DP-NLP model outperforms existing methods in terms of accuracy and privacy by a significant margin. We also run individual ablation experiments to show how the position of the DP layer, sub-sampling ratio, and up-sampling ratio improve privacy and utility, respectively.

Data Sets: Two real-world datasets from the GLUE benchmark (Wang et al., 2018), Quora Question Paris (QQP) and Stanford Sentiment Treebank (SST-2), are used for text-matching and classification tasks. We leave the introduction of these two datasets in the appendix.

For QQP, because each pair of sentences is supplied by the user for training, each pair passes through the DP layer is protected. Given that each pair's binary label contains no privacy, it is sent out without noise perturbation. During the query stage, however, only the DP protected query sentence representation is sent to the server. By contrast, only a single sentence needs to be protected for SST-2, resulting in less calibrated noise according to Equation (7). As a result, it is expected that, in general, better accuracy will be obtained for SST-2 data while maintaining the same privacy budget.

Model and Parameters: Depending on the practical computation and communication resources, we test both the lightweight BiLSTM and BERT model as the encoder and the corresponding model and parameters are provided in the appendix.

Baselines: To protect privacy at the word level, a relaxation of the above (ϵ, δ)-DP definition known as metric DP (Chatzikokolakis et al., 2013) and the corresponding mechanism has recently been proposed (Feyisetan et al., 2020a; Yue et al., 2021; Qu et al., 2021). Because of the unique mechanism used, sampling amplification and tight composition are still absent. As a result, only the DP cost of each training step is given in Feyisetan et al. (2020a); Yue et al. (2021); Qu et al. (2021). To account for the total privacy cost for all the data used, as far as we know, the best way to is to apply the advanced composition (Dwork et al., 2010) to achieve an overall privacy cost. For every $\epsilon > 0, \delta, \delta' > 0$, (ϵ, δ)-DP mechanism is $(\epsilon', T\delta + \delta')$-DP under T-

LDP-NLP Method	DP Definition	DP Granularity	Training/Inference DP Amplification	Training/Inference DP Composition
Lyu et al. (2020)	(ϵ, δ)-DP	feature	✓(with dropout)	✗
Feyisetan et al. (2020a)	metric DP	word	✗	✗
Qu et al. (2021)	metric DP	word	✗	✗
Yue et al. (2021)	metric DP	word	✗	✗
Ours	(ϵ, δ)-DP	sentence	✓(with sampling)	✓

Table 1: Summary of different methods' functionalities for an LDP-NLP task.

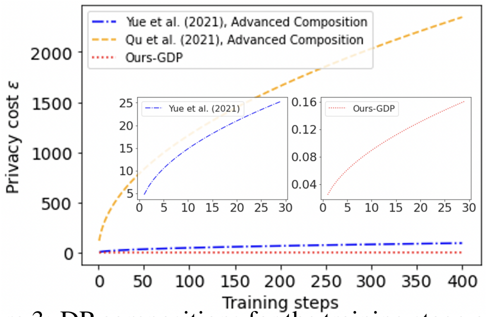


Figure 3: DP compositions for the training stage of various methodologies.

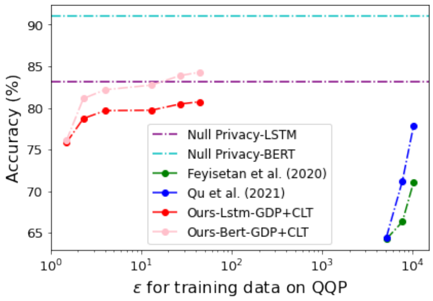


Figure 4: Acc vs. training privacy on QQP.

fold adaptive composition, for

$$\epsilon' = \sqrt{2k \ln\left(1/\delta'\right)} \cdot \epsilon + k \cdot \epsilon \left(e^{\epsilon} - 1\right). \quad (10)$$

Besides, the utility of the null privacy case, which serves as the upper bound, is also provided.

DP Amplification&Composition The ultimate goal is to achieve a reasonable total privacy cost for the entire training and query separately. The total DP cost ϵ, which is a function of training steps, boosts using the advanced composition method (Dwork et al., 2010), as shown in Fig.3. The greatest saving of privacy cost as shown in Fig.3 is due to the fact that the proposed DP layer benefits from the model's inherent randomness, i.e., sub-sampling for training and a novel up-sampling scheme for inference to perform DP amplification as well as a tight privacy composition. Therefore, we can calibrate the noise power in Algorithm 1 tightly so as to improve the model utility as shown in the following.

Training Data Privacy vs. Model Accuracy

We first compare the performance of our proposed method to previous works (Feyisetan et al., 2020a; Yue et al., 2021; Qu et al., 2021) at different DP budget constraints for the entire training data sets. Since none of existing methods consider the privacy protection of inference data, we leave this case in the next subsection. The result in Feyisetan et al. (2020a) is reproduced by Qu et al. (2021), and we select the smallest d for neighbor search as a lower bound from the Fig. 3 in Qu et al. (2021), which gives the strongest privacy. For both QQP and SST-2 data sets, it is shown consistently in Fig. 4 and Fig. 5 that missing a tight DP accounting method results in the total privacy parameter ϵ scaling to more than 5000 in the state-of-the-art works, which does not guarantee any privacy for the whole data set even though they provide reasonable privacy protection for each step. Moreover, due to the random noise applied to the token layer, the performance of existing method degrades significantly compared to the null privacy case and even tend to be that of a random classifier when $\epsilon = 5100$. By contrast, the proposed DP-NLP model improves the performance, which approaches the null privacy case for the SST-2 data sets for both the LSTM and BERT encoders. Moreover, it is observed that the performance loss to the non-DP version of BERT model is larger than that of the LSTM model because its large representation dimension is more sensitive to clipping and noise.

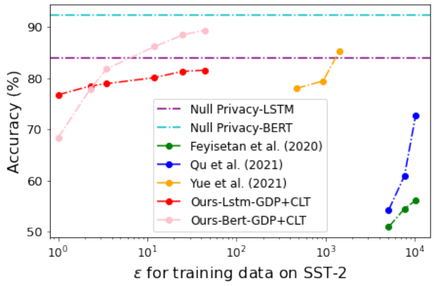


Figure 5: Acc vs. training privacy on SST-2.

Inference data ϵ	Accuracy on QQP Data		
	no USDPA	**+USDPA**	**+USDPA** (Retrain)
1	75.83	**+1.17**	**+3.82**
2.25	78.72	**+0.53**	**+1.29**
4	79.67	**+0.11**	**+0.64**
14	79.82	**+0.10**	**+0.69**
Null Privacy	83.11		

Table 2: Accuracy improvement by up-sampling DP amplification (USDPA) for inference on QQP.

Inference Data Privacy vs. Model Accuracy

We further test the proposed up-sampling DP amplification algorithm to the accuracy improvement. First, we directly apply Algorithm 2 on inference/query data to check the improvement, which is referred to as (up-sampling DP amplification) USDPA. In this case, the noise power for training is not consistent with the training case. We further retrain the model with the same noise power as the inference case to test the performance, which is referred to as USDPA (Retrain). It is shown that the up-sampling improves accuracy, and the stronger the privacy guaranteed the larger an ac-

Inference data ϵ	Accuracy on SST-2 Data		
	no USDPA	**+USDPA**	**+USDPA** (Retrain)
0.8	76.72	**+1.25**	**+3.31**
1.75	78.44	**+0.77**	**+1.72**
2.25	78.90	**+0.80**	**+1.25**
8	80.70	-0.19	**+0.77**
Null Privacy	83.91		

Table 3: Accuracy improvement by up-sampling DP amplification (USDPA) for inference data on SST-2.

Training ϵ	QQP		Training ϵ	SST-2	
	Token Rep.	Latent Rep.		Token Rep.	Latent Rep.
1.5	71.53	**75.83**	1	68.23	**76.72**
2.3	72.85	**78.72**	2.3	71.33	**78.44**
4	74.25	**79.67**	3.5	73.32	**78.90**
13	74.51	**79.82**	12	73.51	**80.70**
45	75.51	**80.47**	25	74.20	**81.30**
Null Privacy	83.11		Null Privacy	83.91	

Table 4: DP layer applied to the token representation versus that applied to the latent representation.

curacy gain is obtained. Note that in practice, the retrain is not necessary and we can directly apply the (up-sampling DP amplification) USDPA.

In the following, we conduct a series of ablation experiments to test the proposed LDP-NLP model.

DP-Layer Position Impact We put the proposed DP layer position to the test by applying it to both the token and latent representations, illustrated in Table 4. Compared to being applied on latent representations, being applied on the token representation is sensitive to random perturbation and results in performance degradation for downstream tasks. Using the DP layer directly on the latent representation, on the other hand, improves accuracy by 4% to 8% for a variety of tasks.

Sub-sampling Rate Impact We also look at how the subsampling rate p_{train} affects model accuracy and training efficiency. The smaller the p_{train}, the smaller the sampled batch size, and thus the slower the convergence. However, according to (8), smaller p_{train} leads to larger DP amplification, resulting in a lower calibrated noise power. As a result, there is a "best" p_{train} to choose from. We fix the privacy budget $\epsilon = 12$ and 45 for SST-2 and QQP to test the effect of p_{train}. The results of Fig. 6 agree with our intuition, and there is a p_{train} that produces the highest accuracy given a fixed privacy budget. As the accuracy is not sensitive to the value of p_{train}, we do not need to tune this parameter in practice to achieve "optimal" performance.

Up-sampling Rate Impact Similar to the above ablation study, we examine the impact of $q \cdot p_{\text{query}}$ for inference. It is expected that the smaller the $q \cdot p_{\text{query}}$ is, the larger accuracy gain we can obtain from the DP amplification. It is evident, however, that we need more sampling times and inference steps to finish all the test datasets, which further increase the privacy cost. Consistent with the above analysis, Figure 7 illustrates the relationships, and in practice, we can specify specific $q \cdot p_{\text{query}}$ values based on query time and privacy budget requirements. The ratio of fictitious data to true data is set to $0, 0.5, 1, 1.5, \ldots, 8.5, 9$, in Figure 7, and $q \cdot p_{\text{query}}$ is set to be the reciprocal of total data size.

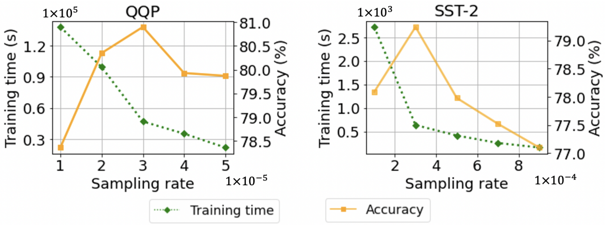


Figure 6: Effectiveness and efficiency relationship influenced by sampling rate for training phrase.

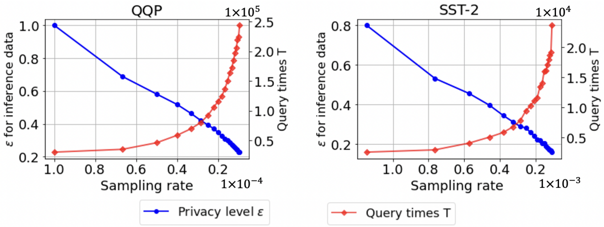


Figure 7: The relationship between privacy cost (total privacy budget, sampling rate $q \cdot p_{\text{query}}$ and the query times to complete all the test sentences.

Moreover, in Appendix A.4, we also discuss the effect of representation dimension on performance. Furthermore, Equations (8) and (9) are used to compute the composition's limits, as explained before Equation (8). It leads to an underestimation of the true cost of privacy. In Appendix A.5, we compute the upper bound further. It is demonstrated that the difference between the lower and upper bounds is very small, implying that the lower bound is a good approximation.

7 Conclusion

To protect the privacy of local user data while keep the model accuracy, we propose a novel LDP-NLP methodology, which includes a non-parametric DP-layer applied to the user-side latent representation, DP amplifications for training/inference data via sub-sampling/up-sampling, tight composition for privacy accounting, and noise calibration algorithms based on DP analysis. It successfully reduces calibrated noise and achieves a significant accuracy improvement while lowering total privacy costs to less than 10 for the first time for both training and inference stages.

References

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*, pages 308–318.

Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. 2021. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. 2020. Deep learning with gaussian differential privacy. *Harvard data science review*, 2020(23).

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

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

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Jinshuo Dong, Aaron Roth, and Weijie Su. 2021. Gaussian differential privacy. *Journal of the Royal Statistical Society*.

Christophe Dupuy, Radhika Arava, Rahul Gupta, and Anna Rumshisky. 2021. An efficient dp-sgd mechanism for large scale nlp models. *arXiv preprint arXiv:2107.14586*.

Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380.

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

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. 2010. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020a. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 178–186.

Oluwaseyi Feyisetan, Sepideh Ghanavati, and Patricia Thaine. 2020b. Workshop on privacy in nlp (privatenlp 2020). In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 903–904.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. 2021. Numerical composition of differential privacy. *arXiv preprint arXiv:2106.02848*.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

Lingjuan Lyu, Xuanli He, and Yitong Li. 2020. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2355–2365.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 257–266.

Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568.

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

Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321.

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

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. 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)*, pages 380–385.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. 2019. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 332–349. IEEE.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. 2021. Differential privacy for text analytics via natural text sanitization. *arXiv preprint arXiv:2106.01221, accepted to ACL-ICJNLP'21 Findings*.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. idlg: Improved deep leakage from gradients. *ArXiv*, abs/2001.02610.

A Supplementary Formalism Details

A.1 GDP Preliminary

GDP is a dual representation of (ϵ, δ)-DP for Gaussian mechanism. Let P and Q denote the distributions of $M(x)$ and $M(x')$ with $x \sim x'$, and let ϕ be any (possibly randomized) rejection rule for testing $H_0 : P$ against $H_1 : Q$. With these in place, Dong et al. (2021) defines the *trade-off function* of P and Q as

$$\begin{aligned} \mathrm{T}(P, Q) &: [0,1] \mapsto [0,1] \\ &\alpha \mapsto \inf_{\phi} \left\{1 - \mathbb{E}_Q[\phi] : \mathbb{E}_P[\phi] \leqslant \alpha\right\}. \end{aligned} \tag{11}$$

Above, $\mathbb{E}_P[\phi]$ and $1 - \mathbb{E}_Q[\phi]$ are type I and type II errors of the rejection rule ϕ, respectively. It is shown that $\mathrm{T}(P, Q) \geq \mathrm{T}(\mathcal{N}(0,1), \mathcal{N}(\mu, 1)) \triangleq G_\mu$, which is referred to as μ-GDP. The conversion between μ-GDP and (ϵ, δ)-DP follows the privacy profile in (5). Please refer to (Dong et al., 2021) for more details about the μ-GDP definition.

A.2 Datasets

Quora Question Paris (QQP): Automated processing of users queries and records is a significant research direction, and one such task is computing the semantic similarity between user logs for the benefit of retrieval efficiency. QQP is a sentence-pair classification task dataset with 363k sentence pairs for training and 40k sentence pairs for validation. The goal is to determine whether a pair of questions are paraphrases or not.
Stanford Sentiment Treebank (SST-2): SST-2 is a sentence classification task that consists of 67k training sentences and 872 validation sentences. The goal is to predict a sentiment label for a movie review sentence. We use the GLUE version of SST-2 (Bowman et al., 2015).

A.3 Models and Parameters

Given the computation and communication cost constraints, we use the lightweight BiLSTM as the user-side feature extractor and set the max sequence length and latent representation dimension to 128. The clipping value is $C = 0.5$, and the DP budget parameters δ for the training/inference stages are set to be the reciprocal of the training/inference data size; and ϵ for the total privacy cost are set to be in the range $[0.8, 14]$. Given a privacy budget of (ϵ, δ), the noise powers for training and inference are calibrated via Algorithm 2 and Algorithm 3.

A.4 Representation Dimension Impact

We investigate the effect of the latent representation dimension on the accuracy for the LSMT model. The clip operation for the sensitivity bound in Algorithm 1 is affected by the size of the ℓ_2-norm of a latent representation. As the norm increases, the clip operation becomes more detrimental to downstream tasks even though we tune it extensively. The outcome demonstrates the critical nature of lower dimension accuracy in the presence of privacy constraints. Additionally, this conclusion is supported by practical constraints on communication and computation.

Dataset	Training ϵ	Latent Rep dimensionality		
		128	256	768
QQP	1.5	**75.83**	75.68	54.58
	2.3	**78.72**	78.18	71.86
	4	**79.67**	79.03	74.27
SST-2	1	**76.72**	75.45	72.94
	2.3	**78.44**	77.29	73.97
	3.5	**78.90**	77.89	76.03

Table 5: Accuracy of different dimensions of latent representation at different privacy levels for training data.

A.5 Privacy Cost Lower Bound and Upper Bound

The equations (8) and (9) are used to compute the composition's approximation, as explained before (8). It leads to an underestimation of the true cost of privacy. In Table 6, we compute the upper bound corresponding to the ϵ used in Fig. 4 and Fig. 5 further based on the very recent work (Gopi et al., 2021). It is demonstrated that the difference between the lower and upper bounds is very small, implying that the lower bound is a good approximation. We do not use the upper bound as the approximation because it is based on numerical computation (Gopi et al., 2021) and it is difficult, if not impossible, to calibrate the noise power.

QQP Privacy Cost ϵ		SST-2 Privacy Cost ϵ	
GDP+CLT	Upper Bound	GDP+CLT	Upper Bound
1.5	1.7	1	1.2
2.3	2.6	2.3	2.7
4	4.5	3.5	4.6
13	13.9	12	13.2

Table 6: Privacy cost based on GDP with CLT (Dong et al., 2021) and compostion of tradeoff functions (Gopi et al., 2021).