Learning and Evaluating a Differentially Private Pre-trained Language Model

Shlomo Hoory,*[*] Amir Feder, Avichai Tendler, Alon Cohen, Sofia Erell, Itay Laish, Hootan Nakhost, Uri Stemmer, Ayelet Benjamini, Avinatan Hassidim and Yossi Matias
Google
Tel Aviv, Israel
{afeder,tendler,aloncohen,rovinsky}@google.com

Abstract

Contextual language models have led to significantly better results, especially when pre-trained on the same data as the downstream task. While this additional pre-training usually improves performance, it can lead to information leakage and therefore risks the privacy of individuals mentioned in the training data. One method to guarantee the privacy of such individuals is to train a differentially-private language model, but this usually comes at the expense of model performance. Also, in the absence of a differentially private vocabulary training, it is not possible to modify the vocabulary to fit the new data, which might further degrade results. In this work we bridge these gaps, and provide guidance to future researchers and practitioners on how to improve privacy while maintaining good model performance. We introduce a novel differentially private word-piece algorithm, which allows training a tailored domain-specific vocabulary while maintaining privacy. We then experiment with entity extraction tasks from clinical notes, and demonstrate how to train a differentially private pre-trained language model (i.e., BERT) with a privacy guarantee of $\epsilon = 1.1$ and with only a small degradation in performance. Finally, as it is hard to tell given a privacy parameter ϵ what was the effect on the trained representation, we present experiments showing that the trained model does not memorize private information.

1 Introduction

Recent advancements in natural language processing (NLP), mainly the introduction of the transformer architecture and contextual language representations, have led to a surge in the performance and applicability of large language models (Vaswani et al., 2017; Devlin et al., 2019). Such models rely on pre-training on massive self-labeled corpora to incorporate knowledge within the language representation. Additionally, when presented with a new dataset and task, such models often gain from an additional pre-training stage, where they are trained to solve a language modeling task on the new training data.

While the pre-training steps are crucial for good model performance on downstream tasks, it can come at the expense of the privacy of the persons mentioned in the data. As these models learn to predict words using their context, they often memorize individual words and phrases. Such memorization can lead to information leakage when using the trained models or the language representation. This problem is particularly acute in medical domains, where sensitive patient data might leak (Hartman et al., 2020; Feder et al., 2020).

One solution for pre-training the model while preserving patients' privacy is to train the model with a differential privacy guarantee (Abadi et al., 2016b). Such guarantee is achieved through a training process which introduces random noise, allowing the modeler to bound the effect an individual has on the model. However, for a sufficiently small privacy parameter ϵ, this usually comes at the expense of model performance. Also, differentially private training schemes were only shown to work for recurrent language models, and not for more recent systems that are based on the transformer architecture (McMahan et al., 2018; Kerrigan et al., 2020).

Apart from their size (110M trainable parameters for BERT), transformer-based language models introduce an additional privacy concern. When using pre-trained language models on new datasets, we can often improve performance by learning a new domain-specific vocabulary, and re-training the model with the new tokenizer (Section 5). Unfortunately, commonly used transformer-based models such as BERT rely on the WordPiece tokenization algorithm (Wu et al., 2016b), which uses the distribution of words in the data and can therefore

[*]Tel-Hai College, Israel. Work was done while at Google.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1178–1189
November 7–11, 2021.

potentially leak private information as well.

Finally, even if we successfully train a contextual embedding model with a sufficiently small ϵ guarantee, it is hard to test and evaluate the resulting privacy-preserving properties of the model. One also has difficulty understanding whether the differentially-private training procedure affected the language representation other than by measuring performance on a downstream task. For example, it could be that other valuable information was also lost during the differentially private training.

In this work, we provide a detailed solution to training a differentially-private vocabulary and contextual embedding model, and to better understanding the resulting representation. We present a method for training BERT, a contextual embedding model, on medical data with a strong privacy guarantee of $\epsilon = 1.1$ in total (including the private WordPiece and pre-training algorithms) and with only a small degradation in performance (Section 2.1). To do that, we introduce the first differentially private WordPiece algorithm, designed to generate a new domain-specific vocabulary while maintaining user privacy (Section 3.2). Following that, we successfully generate a differentially private BERT model, which uses the new vocabulary to improve results on the downstream tasks.

Possibly the most major technical challenge in pre-training a differentially-private contextual embedding model is the fact that the training batch size has to be very large ($128K$), all the while training on specific hardware (TPUs) in which the batch size is limited. We overcome this obstacle by spreading each training batch over time during the training process, along with other useful manipulations we discuss in Section 2.1. Finally, after training the differentially-private BERT on clinical notes, we follow common wisdom (Carlini et al., 2019) and provide privacy tests showing that information leakage has been prevented in this process (Section 5). We hope that this work will further improve user privacy, and will spur more theoretical and empirical research in the intersection of differential privacy and natural language processing.

2 Previous Work

Since the introduction of the differentially-private Stochastic Gradient Descent (SGD) algorithm (Song et al., 2013; Abadi et al., 2016b), it is possible to train deep neural networks (DNN) with privacy guarantees. Specifically, there have been several attempts to train DNN-based language models with such guarantees, though with mixed results in terms of performance on downstream tasks (McMahan et al., 2018; Kerrigan et al., 2020). To better understand the trade-offs between the performance and privacy of deep language models, we survey here the literature on differentially-private training and on methods for measuring privacy in language models.

2.1 Training Differentially-Private Models

Differential Privacy (DP; Dwork et al., 2006; Dwork, 2011; Dwork et al., 2014) is a framework that quantifies the privacy leaked by some randomized algorithm accessing a private dataset, reader unfamiliar with DP, can consult the short introduction in Appendix A. In the context of training a machine learning model on private data, it enables one to bound the potential privacy leakage when deploying the model to the world.

Definition 1 ((ϵ, δ)-DP)**.** *Given some $\epsilon, \delta > 0$, we say that algorithm $\mathcal{A}$ has (ϵ, δ)-differential privacy, if for any two datasets D, D' differing in a single element and for all $S \subseteq Range(\mathcal{A})$, we have:*

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

The leading method for training models with small differential privacy parameters ϵ, δ is the DP-SGD method introduced by Abadi et al. (2016b). The method was subsequently incorporated into Tensorflow's privacy toolbox with improved privacy analysis (Mironov, 2017; Mironov et al., 2019). The basic idea behind DP-SGD is to clip and add noise to the per-example gradients of the loss function during model training. The intuition is that such a mechanism guarantees that, for each step, the influence of each example on the outcome is bounded.

In the context of NLP, there have been several attempts to train language models using the DP-SGD algorithm. Specifically, McMahan et al. (2018) presented a pipeline for training differentially-private language models based on the recurrent neural network (RNN) architecture. While successful on the RNN architecture, results on a fine-tuned transformer, specifically GPT-2, were shown to be less successful in preserving privacy without hurting task performance (Kerrigan et al., 2020). In this paper, we present the first, as far as we know, successfully trained differentially private BERT model, with a strong privacy guarantee and with only a small decrease in downstream performance.

2.2 Evaluating the Privacy of Language Models

While differential privacy training provides privacy guarantees (in terms of the privacy parameters ϵ, δ), it is often hard to evaluate the practical implication of such a guarantee. When evaluating language models it becomes even trickier, as private information might be encoded in specific phrases contained in the text, but it can also be implicitly contained in the language model. In the context of clinical notes, for example, information regarding the linguistic style of the doctor can be captured and predicted from linguistic cues in the text itself (Rosenthal and McKeown, 2011; Preoţiuc-Pietro et al., 2015; Coavoux et al., 2018).

Song and Raghunathan (2020) studied information leakage from language representations, and presented several methods for evaluating the privacy preserving qualities of trained language models. They provided a taxonomy of adversarial attacks, differing by the adversary's access to model's internal state. Specifically, they defined membership attacks on language representation, which are designed to detect memorized information. In this paper, we build on the *secret sharer* membership test, a method for quantitatively assessing the risk that rare or unique training-data sequences are unintentionally memorized by generative sequence models (Carlini et al., 2019). While not specifically designed for language models such as BERT, it fits the DP evaluation setup perfectly. Concretely, in this test a secret sharer plants n identical occurrences of a k-WordPiece token sequence into the train corpus. The sequence itself consists of i.i.d. random tokens where the secret is the middle token. The model is then trained on the modified corpus and evaluated for each planted sequence by trying to predict the secret token.

In Section 5, we show that unlike the original BERT model, our trained DP-BERT model does not memorize sequences of words introduced via the secret sharer.

3 Training Differentially Private Contextual Language Models

Training differentially private language models becomes exceedingly difficult with model size. Hence, attempting to train a transformer model such as BERT using the DP-SGD algorithm, without any modifications, is bound to result in a significant performance degradation (Kerrigan et al., 2020). Moreover, as the WordPiece algorithm, the process that tokenizes the textual input of BERT, is not differentially private, re-training it to fit a domain-specific vocabulary will not guarantee that there is no information leakage regardless of the DP-SGD training. In this section, we formulate the problem of training a DP BERT model on medical text, and explain the process of constructing a differentially private vocabulary. We then discuss the importance of parallel training and very large batch sizes in training such large language models, and provide a method for sufficiently increasing such crucial parameters.

3.1 Problem Formulation

We choose to focus our DP training on entity extraction (EE) tasks from medical text, specifically clinical notes. Clinical notes include medically relevant information regarding patients' conditions, and are often used as training data for downstream machine learning tasks (Esteva et al., 2019). However, they can contain private information that might put patients at risk (Feder et al., 2020; Hartman et al., 2020). For this reason, language models trained on such datasets must be able to learn domain-relevant information (such as medical jargon and doctors' writing style) without memorizing private information (Lee et al., 2020).

To test our ability to train a DP language model on clinical notes, we use a BERT model (Devlin et al., 2019) with specialization to the medical domain. To this end, the public Wikipedia and BookCorpus datasets (Zhu et al., 2015) used to train BERT were amended with the Medical Information Mart for Intensive Care III corpus (Johnson et al., 2016, MIMIC-III) in order to improve performance on medical tasks.

Before introducing changes designed to guarantee privacy, let us review the procedure used to obtain the Medical BERT model. The available resources are the 3 billion word Wikipedia + BookCorpus datasets, and the 712M word MIMIC-III corpus. The training process consists of the following three steps:

(i) Build the vocabulary from the MIMIC-III corpus.

(ii) Train BERT from scratch on the Wikipedia + BookCorpus using the new vocabulary.

(iii) Continue BERT's training on the MIMIC-III corpus.

The steps that are susceptible to leaking MIMIC-III data are the first, and the third. Therefore, by the composability property of differential privacy (Dwork et al., 2014, Theorem 3.16), our problem reduces to providing algorithms with satisfactory DP guarantees for steps (i) and (iii) without causing a significant performance loss. We discuss these problems in detail in the following two subsections.

3.2 Constructing a differentially private vocabulary

Transformer-based models commonly tokenize inputs into sub-words using the WordPiece algorithm. The WordPiece algorithm (Wu et al., 2016a) is a general method for improving the generalization properties of a language model by tokenizing based on the most frequent combination of symbols rather than words. While its efficacy is undisputed, it can leak private information by memorizing certain tokens in the training data. To prevent such leakage, we modify this algorithm to satisfy DP. We do so by introducing noise to the word histogram used in its training process.

The WordPiece algorithm starts with constructing the word histogram of the corpus. This histogram is then manipulated to obtain the WordPiece output vocabulary through an iterative process which forms sub-words according to their likelihood. Since DP is robust to post-processing, making the input histogram DP is sufficient to guarantee a DP end-result vocabulary (Dwork et al., 2006). Our DP WordPiece algorithm therefore adds noise to the histogram with given privacy parameters and then applies the standard WordPiece algorithm.

There exist techniques to generate histograms with differential privacy, e.g. Korolova et al. (2009) and (Bun et al., 2019). The situation encountered in language models is slightly different, since we wish to protect not the privacy of a single word in the histogram, but of a larger entity such as an example spanning many words. In this work we guarantee differential privacy at the level of a single training example, $N = 256$ words, to be consistent with the differential privacy guarantee by the training process itself.

Given a textual dataset over the set of words X, we partition the dataset into a sequence D of N-word tuples. For each tuple v, we define its word histogram $f_v : X \to \mathbb{R}$ as:

$$f_v(x) = \begin{cases} 1, & \textit{if } x \in \textit{Supp}(v) \\ 0, & \textit{otherwise} \end{cases}$$

Note that this is not exactly the word histogram of the text, since each distinct word is counted exactly once, regardless of the number of times it appeared in the tuple. This heuristic is useful to get a better DP bound and describe below. It can possibly reduce utility and somewhat change the vocab obtained, since it is not the exact word histogram.

We use f_v to construct an (ϵ, δ)-DP histogram h using the procedure described next. One should also note that the construction holds for a general f_v, not necessarily the one defined above.

Given a collection of datasets $\mathcal{D}$, where each dataset $D \in \mathcal{D}$ is a sequence of tuples in X^N, a function $f : X^N \to \mathbb{R}^X$, and some constants $C, \sigma > 0$, we define a randomized function $h : \mathcal{D} \to \mathbb{R}^X$ by the following process:

1. Set $h'(D) = \sum_{v \in D} f(v)$.
2. For all coordinates $x \in \textit{Supp}(h')$ add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the x coordinate in $h'(D)$.
3. Clip h' as follows to get h:

$$h(D) = \begin{cases} h'(D), & \textit{for } h'(D) \geq C \\ 0, & \textit{otherwise} \end{cases}$$

Now, using the above definitions, we can prove that our newly modified WordPiece algorithm is indeed differentially private. Specifically, the following theorem holds:

Theorem 1. *With the notations above, let* $k, m, \delta > 0$, $\epsilon = \frac{k}{\sigma}\sqrt{2\log(2.5/\delta)}$ *and* $C = m + \sigma \operatorname{erf}^{-1}(1 - \delta/2N)$.

Then, if $\|f(v)\|_2 < k$, $\|f(v)\|_\infty < m$, *and* $\operatorname{supp}(f(v)) \subset \operatorname{supp}(v)$ *hold for all* $v \in X^N$, *then* h *is* (ϵ, δ)*-DP.*

Proof. Given two neighboring datasets $D, D' = D \cup \{v\}$ where $v \in X^N$. We divide the coordinates of v into two sets:

For elements in the vector v which already appear somewhere in D, the construction of h' is just the Gaussian mechanism because the L2-norm bound on f, which is $(\epsilon, \delta/2)$-DP as shown in (Dwork et al., 2014). Therefore h is also $(\epsilon, \delta/2)$-DP, as post processing of h'.

If x is an element of X that appears in v but not in D, then $h'(D)(x) = 0$ and

$$h'(D')(x) = f(v)(x) + \mathcal{N}(0, \sigma^2) < m + \mathcal{N}(0, \sigma)$$

Where in the last inequality we used the bound on $\|f(v)\|_\infty$. This will be clipped unless $h'(D')(x) > C = m+\sigma\,\mathrm{erf}^{-1}(1-\delta/2N)$ the probability of which is smaller then $Pr[\mathcal{N}(0,\sigma^2) > \mathrm{erf}^{-1}(1-\delta/2N)] = \delta/2N$. By the requirement on the support of f, there are at most N such coordinates x, so by the union bound we get a non-zero with probability $< \delta/2$. Which makes this part $(0, \delta/2)$-DP.

So, summing up the contribution of both parts we get (ϵ, δ)-differential privacy. □

We apply the theorem with $m = 1$ as f is at most 1 in all coordinates, and with $k = \sqrt{N}$, as the maximal L2-norm is obtained for tuples of N-distinct words and $\|(1,\ldots,1)\|_2 = \sqrt{N}$. In this work we used $N = 256$.

Corollary 1. *Let $\sigma > 0, 0 < \delta < 1.25e^{-3/2}$, and let $C = 1 + \sigma\,\mathrm{erf}^{-1}(1-\delta/N)$. Then, the above procedure yields an (ϵ, δ)-DP histogram h, with $\epsilon = \frac{\sqrt{N}}{\sigma}\sqrt{2\log(1.25/\delta)}$.*

The theorem proves the corollary with slightly worse bounds: $C = 1 + \sigma\,\mathrm{erf}^{-1}(1-\delta/2N)$ and $\epsilon = \frac{\sqrt{N}}{\sigma}\sqrt{2\log(2.5/\delta)}$. For the proof of the corollary as stated, which allows us to decrease ϵ and therefore improve the privacy guarantee, Appendix B.

Parameters for learning a DP-vocabulary In this work we used corollary 1 with $N = 256$ and required $\delta = 10^{-9}$. We added a noise with $\sigma = 200$, as in the corollary we used $C = 982.5$ and obtained $\epsilon = 0.517$ (denoted as ϵ_V in Section 5). We applied WordPiece on the DP-histogram, the resulting vocabulary had 20,855 WordPieces, compared to 29,157 when WordPiece was applied to the original histogram.

3.3 Training a differentially private BERT

Equipped with a DP trained vocabulary, we can now train our language model. To train a differentially private contextual embedding model (i.e. BERT), we use the DP-SGD method supplied by the TF privacy toolbox (see Section 2.1). The parameters of the algorithm are the number of steps, batch-size B, ℓ_2-norm-clip C, and the noise multiplier σ. To fix notation, we formally define the DP-SGD step, as defined in Abadi et al. (2016b, Algorithm 1). Given the per-example gradients of the loss function $g_1,\ldots,g_B$, the gradient $\tilde{g}$ for passing to `apply_gradients` is defined by:

$$\overline{g_i} = g_i / \max(1, \|g_i\|_2/C), \quad \text{for all } i; \tag{1}$$

$$\tilde{g} = \frac{1}{B}\left(\sum_i \overline{g_i} + \mathcal{N}(0, \sigma^2 C^2 \mathbb{I})\right). \tag{2}$$

The most important parameter of the algorithm is the noise multiplier σ – increasing σ directly decreases ϵ; i.e., increases the differential-privacy guarantee of the algorithm. On the other hand, it harms performance on the target data-set, and thus a careful choice of σ is necessary to balance the trade-off between privacy and performance. We choose the noise σ to be proportional to the square root of the batch size B. This is done in order to make the privacy guarantee oblivious to changes in the batch size B (as one can observe from Eq. (2)). The privacy guarantee is also affected by the number of training steps (or epochs), but this behavior is more gradual since ϵ increases near-linearly in the range of interest. In our experience, the clip level C is of lesser importance and we fix it to be 0.01.

For any choice of parameters, we upper bound the privacy parameter ϵ using the TF privacy toolbox `compute_dp_sgd_privacy` function, where we also use the number of MIMIC examples $N = 83M$. We fix privacy δ to be 10^{-8}, which is smaller than $1/N$.

The effect of parallelism. In order to make the training run faster, we use TPUs[1] to parallelize training by splitting example batches to shards. This mechanism is readily available through Tensorflow (TF; Abadi et al., 2016a), but its effect has to be taken into account when computing the bounds on ϵ.

In order to understand this effect, let us first review the way we incorporate TF privacy into the BERT training procedure. The change consists of changing the loss computation code to compute the vector loss (per-example loss), and of wrapping the existing Adam weight decay optimizer (Kingma and Ba, 2015), our optimizer of choice, by the DP optimizer using the `make_gaussian_optimizer_class` method.

The subtle point lies in the second change, as the optimization is also wrapped by `CrossShardOptimizer` which handles

[1] https://cloud.google.com/tpu/docs/tpus.

the sharded batching. Let B denote the unsharded batch size, and P denote the number of parallel shards. For each batch, the examples are split between P independent instances of the TF privacy optimizer, each handling B/P examples. For each shard, the gradients are clipped, averaged and noise is added by equations Eqs. (1) and (2). Subsequently, the `CrossShardOptimizer` averages the P shard gradients to obtain the single gradient to be passed to `apply_gradients`.

Therefore, denoting the i-th gradient of shard j by $g_{i,j}$, the gradient passed to `apply_gradients` can be written as follows:

$$\tilde{g} = \frac{1}{P}\sum_j \left[\frac{1}{B/P}\left(\sum_i \overline{g_{i,j}} + \mathcal{N}(0,\sigma^2 C^2 \mathbb{I})\right)\right] = \frac{1}{B}\left(\sum_{i,j} \overline{g_{i,j}} + \mathcal{N}(0, P\sigma^2 C^2 \mathbb{I})\right). \quad (3)$$

This implies that using noise multiplier σ with P shards is equivalent to an unsharded training with noise multiplier $\sigma\sqrt{P}$. As computing an upper bound on ϵ through TF privacy does not take parallelism into account, one must use $\sigma\sqrt{P}$ as the noise multiplier in order to get the correct result.

Achieving larger batch sizes. As it quickly became apparent, to successfully train a large transformer with DP-SGD, larger batch sizes are required. However, usually batch size cannot increase beyond a certain point because of memory considerations and limitation on the number of available TPUs. With the resources available to us, for example, we couldn't get beyond parallelism of $P = 256$ with sharded batch size of 32, achieving total batch size $B = 8192$.

We chose to solve this problem by spreading the batch in time, so `apply_gradients` is called only once every T batches with the total average gradient. This is equivalent to increasing both P and B by a factor of T. With this method, the only limit on T is processing time. From our experience, the value of $T = 32$ is a reasonable choice, achieving parallelism of $P = 256 \cdot 32$ and total batch size B of 128k with the above parameters.

We briefly remark upon the implementation of this mechanism. For every trainable variable, we created a variable with `/grad_acc` suffix added to the original name. For each step, the `train_op` either accumulates the current gradients in the new variables, or zeros the accumulator and calls `apply_gradients`, depending on the current step modulo T.

4 Experimental Setup

We design our experiments to demonstrate the ability of the DP training scheme to achieve similar results to the non-DP training scheme on the same data. We focus on the medical domain as it has strict privacy requirements and its language is distinct enough so that additional pre-training should be useful. We start by describing the data used for the DP training and relevant implementation details. We then present the entity extraction task used for the supervised task training and evaluation. Finally, we discuss the relevant baselines, chosen to demonstrate the efficacy of the DP training scheme.

Pre-training data. For the DP pre-training, we supplement the original training data used in Devlin et al. (2019) with the MIMIC-III dataset, a commonly used collection of medical information that contains more than 2 million distinct notes (Johnson et al., 2016; Alsentzer et al., 2019). MIMIC-III covers 38,597 distinct adult patients and 49,785 hospital admissions between 2001 and 2012. The clinical notes in this dataset are widely used by NLP researchers for a variety of clinically-related tasks (Feder et al., 2020; Hartman et al., 2020), and were previously used for pre-training BERT models specifically for the medical domain (Alsentzer et al., 2019).

Using the combined dataset, we train our DP-BERT model using the training scheme described in Section 3.

Entity-extraction task. For the supervised task training, we used two datasets from the i2b2 National Center for Biomedical Computing for the NLP Shared Tasks Challenges: i2b2-2010 and i2b2-2011 (Uzuner et al., 2011). These datasets contain clinical notes tagged for concepts, assertions, and relations (i2b2-2010 - 170 clinical notes, i2b2-2011 - 424 clinical notes). In this task, patient reports are labeled with three concepts: test, treatment, and problem. The total number of entities in each category can be seen in Table 1.

The i2b2-2011 data is split to training (251 notes) and test (173 notes) sets. On i2b2-2010, we perform 5-fold cross validation where each fold has random training (136 notes) and test (34 notes) sets.

Concept	i2b2-2010	i2b2-2011
Problem	7,073	11,924
Test	4,608	8,071
Treatment	4,844	8,328

Table 1: Number of Concept entities included in the i2b2 2010 and 2011 datasets, for each type of Concept (Problem, Test and Treatment).

Baselines. We compare our differentially private BERT model, denoted as BERT-DP, to several non private baselines:

BERT (Wikipedia + Books) We train a BERT-large model, as in Devlin et al. (2019), using the default hyperparameters.

BERT-M (Wikipedia + Books + MIMIC-III) We supplement the original training from Devlin et al. (2019) with the MIMIC-III clinical notes corpus. In addition, we also use a (non-differentially private) WordPiece vocabulary generated from MIMIC-III.

BioBERT We use the training data presented in Lee et al. (2020), and use it to train BERT. We tested version v1.1 which it trained using the original dataset + 1M PubMed abstracts.

In Section 5 we compare several differentially private models, discuss their differences and highlight the effect of certain parameters (as discussed in Section 3) on the EE task performance.

5 Results

In this section we empirically evaluate the trade-offs between a model's privacy and its usefulness. Previously, in Section 3, we have shown how to pre-train a contextual embedding model such as BERT with any, possibly substantial, privacy guarantee. We naturally expect that a stronger privacy guarantee would entail that less information is preserved during pre-training, which in turn would degrade performance on downstream tasks. Thus, we aim to ascertain the exact trade-off between these two goals in order to be able to choose a model that has both good performance and a satisfactory privacy guarantee.

We provide two sets of experiments to help better understand this trade-off as well as to provide practitioners with tools to understand the effects of DP pre-training. First, we use the pre-trained DP model and fine-tune it on the entity extraction task on both i2b2-2010 and i2b2-2011, demonstrating the ability of the differentially private language model to benefit from the pre-training step. Then, we test the ability of the model to memorize private information and show that it is protected against commonly used privacy attacks. Aggregating both results, we argue that medically-relevant information is preserved in the DP model all the while private information is not revealed.

For all our model variants, unless explicitly stated otherwise, the parameters are as discussed in Section 3, with batch size $B = 128k$, noise multiplier $\sigma = 2.72$, and 1M training steps.

5.1 Preserving Useful Information

For our first experiment, we pre-trained a DP BERT model, and then fine-tuned it on an EE task over the i2b2-2010 and i2b2-2011 datasets. We summarize our results in Table 2. As can be seen in the table, the additional pre-training either on MIMIC-III (BERT-M) or on PubMed (BioBERT) gives a significant boost in performance over the off-the-shelf BERT, increasing F1 performance from 76.3 to 86.8 on i2b2-2010 and from 77.6 to 83.6 on i2b2-2011. Importantly, we observe that adding differential privacy guaranties, using the hyperparameters and training procedure discussed in Section 3, degrades performance only slightly. Still, as expected, F1 performance decreases as privacy guaranties improve (ϵ gets smaller), decreasing by 0.8 and 0.5 (in absolute terms) in F1 performance on i2b2-2010 and i2b2-2011, respectively. Indeed, the BERT-DP with the smallest epsilon ($\epsilon_V + \epsilon_T = 1.1$) improves performance by 7.4 on i2b2-2010 and 3.5 on i2b2-2011 (in absolute terms).

Model	ϵ_V	ϵ_T	i2b2-2010	i2b2-2011
BERT	∞	∞	76.3	77.6
BERT-M	∞	∞	86.8	83.6
BioBERT	∞	∞	86.5	–
BERT-DP	0.51	2.8	84.5	81.7
BERT-DP	**0.51**	**0.6**	**83.7**	**81.2**

Table 2: Results on the Medical Entity Extraction task on both the i2b2-2010 and i2b2-2011 datasets. $\epsilon = \infty$ denotes no differential privacy guarantee. BERT-DP with the best privacy guarantee ($\epsilon = 1.1$) highlighted in bold.

In addition, in Fig. 1 we evaluate the change in the DP-SGD $\epsilon = \epsilon_T$ and in the F1 score of the downstream task as a function of the batch size, the noise multiplier σ, and the number of pre-training epochs. The behavior in all three parameters is as expected. Specifically, increasing σ enables more

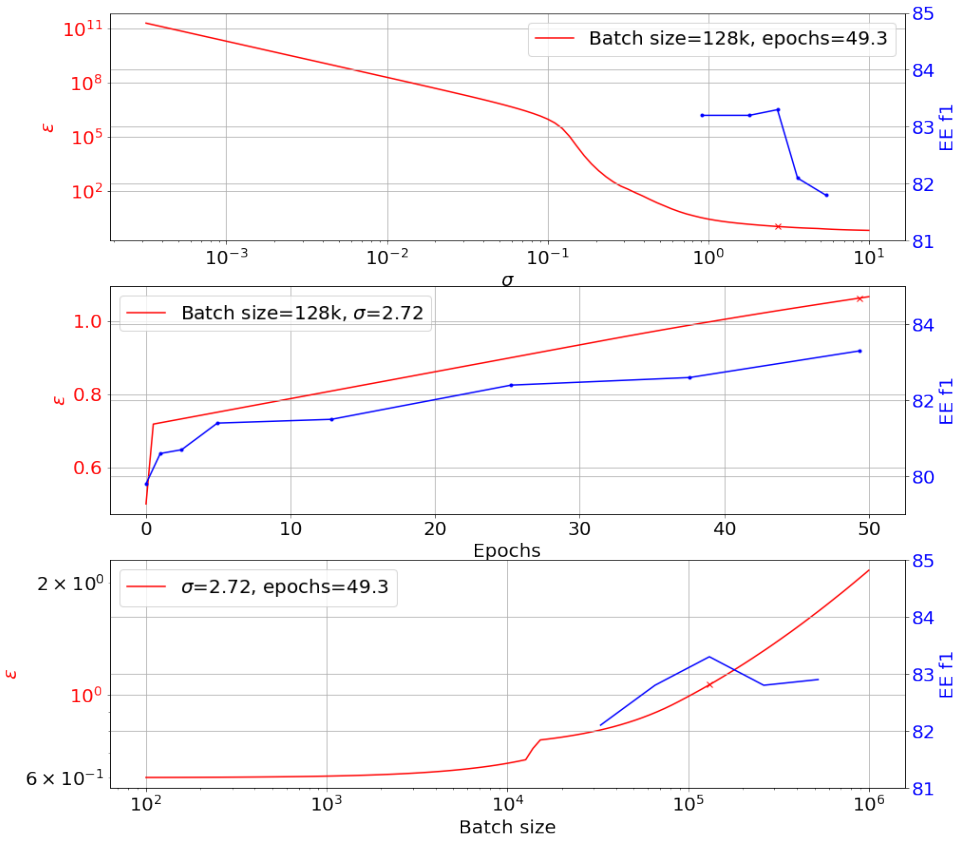

Figure 1: Top to bottom - DP-SGD privacy parameter ϵ (red) and test F1 score on the i2b2-2010 EE task (blue), as a function of: noise multiplier σ; number of pre-training epochs; pre-training batch size.

privacy (lower ϵ), but worsens performance. Similarly, with more pre-training epochs, the model gathers more information about the training data, and so obtain better F1 score but worse privacy preservation (higher ϵ). When increasing the batch size without modifying the noise multiplier σ proportionally, both ϵ and the F1 increase. To summarize, based on the results in Fig. 1, we recommend practitioners interested in generating DP models to opt for very large batch sizes and train for as many epochs as their target ϵ allows them.

5.2 Forgetting Private Information

For our second batch of experiments, we follow Carlini et al. (2019) to test the model's ability to memorize private information. We inject the MIMIC-III data set with "canaries", where canary $C_{k,p}$ is a length k sequence of random word pieces that is injected into a random location for each training example with probability p. For each canary, one word piece is regarded as the secret, while the others as hints. We evaluate a model trained on the injected MIMIC-III data set on the same training examples while masking the secret and using the masked language model task to evaluate the true secret rank. We measure how well the model memorizes the secret by the exposure metric defined as $\log_2(|\text{vocabl}|) - \log_2(\text{average secret rank})$.

We tested the *HS*, *HSH*, and *HHSHH* canary hint/secret patterns for different values of p on a DP model and a non-DP model. As can be seen from Fig. 2, even when the secret appears as much as 100K times in the data, the DP model performs significantly better than the non-DP model. This suggests that the model learns through information that helps it generalize rather than memorize the dataset in its entirety, which includes private and personal information as well.

6 Discussion and Future Work

In this paper, we have shown a procedure for learning and evaluating a differentially-private contextual language model. We have defined the problem of learning such a model with end-to-end privacy guarantees and have discussed the pitfalls that might lead to poor downstream performance. Allowing for vocabulary modifications, we have introduced a novel WordPiece algorithm and proved

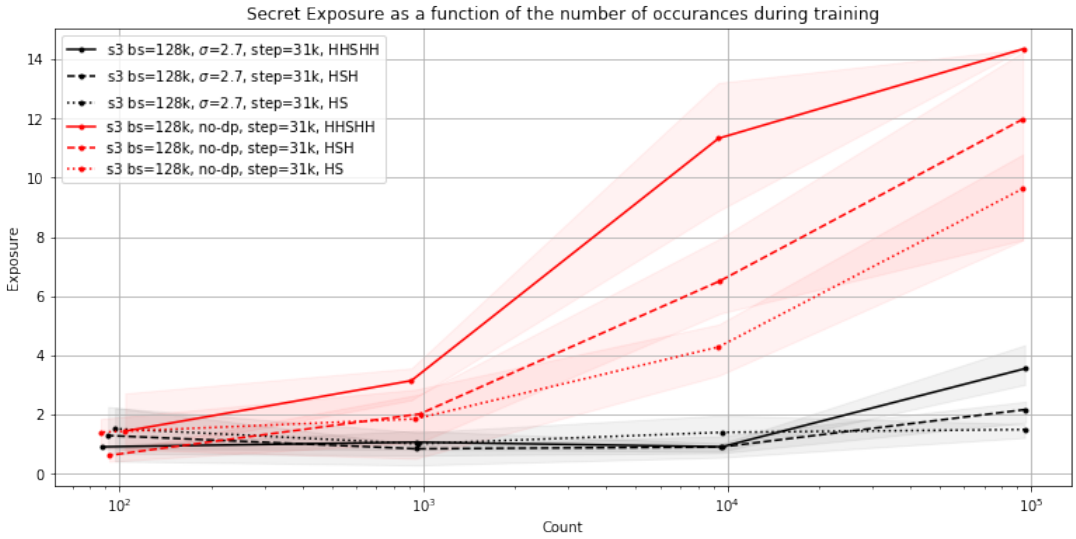

Figure 2: Secret exposure as a function of the number of secret occurrences. Black lines denote models with DP $\epsilon = 1$, red lines models without DP $\epsilon = \infty$. *bs* denotes batch size, *step* denotes step size and *HHSHH/HSH/HS* the hint pattern used.

that it is differentially-private. Then, to overcome the difficulties associated with learning DP contextual language models, we have offered practical measures for circumventing them, most notably through vastly increasing batch sizes. Finally, to increase the trust of the DP trained contextual language model, we have utilized a secret sharer evaluation test and showed that our trained language model does not memorize private information.

While these results are definitely encouraging, more research is needed. Our results are confined to the medical domain, where privacy needs are perhaps most stringent. Showing the efficacy of this training and evaluation pipeline on other domains would certainly increase the trust in it. Additionally, we have not fully explored potentially tighter bounds on our DP WordPiece algorithm. In future work, we plan to provide more theoretical and empirical support for end-to-end privacy guarantees.

Finally, the observed performance gain due to the vocabulary training presents an interesting question for the larger NLP community. Understanding the importance of vocabulary vs. linguistic style when performing additional pre-training could improve the domain adaptation capabilities of existing NLP systems. In future work, we plan to expand our DP training to additional domains, allowing us to test the power of vocabulary modifications via the DP WordPiece training in increasing across domain performance.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016a. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016b. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, WA Redmond, and Matthew BA McDermott. 2019. Publicly available clinical bert embeddings. *NAACL HLT 2019*, page 72.

Mark Bun, Kobbi Nissim, and Uri Stemmer. 2019. Simultaneous private learning of multiple concepts. *J. Mach. Learn. Res.*, 20:94–1.

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.

Maximin Coavoux, Shashi Narayan, and Shay B Cohen. 2018. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cynthia Dwork. 2011. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95.

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, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. 2019. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29.

Amir Feder, Danny Vainstein, Roni Rosenfeld, Tzvika Hartman, Avinatan Hassidim, and Yossi Matias. 2020. Active deep learning to detect demographic traits in free-form clinical notes. *Journal of Biomedical Informatics*, 107:103436.

Tzvika Hartman, Michael D Howell, Jeff Dean, Shlomo Hoory, Ronit Slyper, Itay Laish, Oren Gilon, Danny Vainstein, Greg Corrado, Katherine Chou, et al. 2020. Customization scenarios for de-identification of clinical notes. *BMC medical informatics and decision making*, 20(1):1–9.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Gavin Kerrigan, Dylan Slack, and Jens Tuyls. 2020. Differentially private language models benefit from public pre-training. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 39–45.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. 2009. Releasing search queries and clicks privately. In *WWW*, pages 171–180. ACM.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *International Conference on Learning Representations*.

Ilya Mironov. 2017. R'enyi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE.

Ilya Mironov, Kunal Talwar, and Li Zhang. 2019. R'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*.

Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. 2015. An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1754–1764.

Sara Rosenthal and Kathleen McKeown. 2011. Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 763–772.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 377–390.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, and J. Klingner. 2016a. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016b. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yukun Zhu, Ryan Kiros, Richard S Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*.

A A short introduction to differential privacy.

The results in this section are standard in differential privacy (DP), and will be stated without proofs, for more details and proofs consult e.g. Dwork et al., 2006; Dwork, 2011; Dwork et al., 2014).

A.1 Definition of DP

Suppose we have a dataset which holds private information about individuals. We wish to obtain some information about the dataset, for example descriptive statistics without revealing private information about the individuals.

Not revealing private information is informal, DP formalizes this concept by requiring that adding or removing any individual from a dataset, will not change significantly any probability computed from the information provided. The word "probability" suggests that DP makes sense only in the framework of randomized algorithms, that is why typically in DP one adds noise to the algorithms. The formal definition of DP goes as follows:

Definition 2 (ϵ-DP)**.** *Given $\epsilon > 0$, we say that the randomized algorithm $\mathcal{A}$ has ϵ-DP, if for any two datasets D, D' differing in a single element and for all $S \subseteq Range(\mathcal{A})$, we have:*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in S].$$

In other words, the ratio between the probabilities to obtain some result with or without the individual, is bounded by a small factor. This definition is sometimes too strict, because the condition needs to be satisfied even on very rare events. Most DP papers, including this one, works with an approximate DP definition which allows the above definition to fail with small probability δ, more formally:

Definition 3 ((ϵ, δ)-DP)**.** *Given $\epsilon, \delta > 0$, we say that the randomized algorithm $\mathcal{A}$ has (ϵ, δ)-DP, if for any two datasets D, D' differing in a single element and for all $S \subseteq Range(\mathcal{A})$, we have:*

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

Typically δ is taken smaller than 1/dataset-size, to avoid failures of the definition.

A.2 Example: counting with DP.

Suppose we have a group D of people and we wish find how many of them has a disease, we define $p(x) = 1$ if the person x has the disease and $p(x) = 0$ otherwise.

One method is to release the count directly:

$$\mathcal{A}(D) = \sum_{x \in D} p(x)$$

This method is not DP. Indeed, suppose $\mathcal{A}(D) = N$, or $Pr[\mathcal{A}(D) = N] = 1$. Let D' be obtained by adding a person with the disease to D, then $\mathcal{A}(D') = N + 1$, or or $Pr[\mathcal{A}(D) = N] = 0$. Hence for any $\epsilon > 0$:

$$1 = \Pr[\mathcal{A}(D) \in \{N\}] > e^{\epsilon} \Pr[\mathcal{A}(D') \in \{N\}] = 0$$

And the DP definition is not satisfied.

To obtain a DP version of this count, we can use the Laplace mechanism. Consider the Laplace distribution with density function: $Lap_b(x) = \frac{1}{2b} exp(\frac{|x|}{b})$ We can add noise to the above algorithm making it ϵ-DP as follows:

$$\mathcal{A}'(D) = \mathcal{A}(D) + Lap_{1/\epsilon}(x) = \sum_{x \in D} p(x) + Lap_{1/\epsilon}(x)$$

Indeed for all D, D', x and since $|\mathcal{A}(D) - \mathcal{A}(D')| \leq 1$:

$$\frac{Pr[A'(D) = x]}{Pr[A'(D') = x]} = \frac{Pr[A(D) + Lap_{1/\epsilon} + = x]}{Pr[A(D') + Lap_{1/\epsilon} = x]} =$$

$$= \frac{\frac{\epsilon}{2} exp(\epsilon |x - A(D)|)}{\frac{\epsilon}{2} exp(\epsilon |x - A(D')|} =$$

$$= exp(\epsilon(|x - A(D)| - |x - A(D')|)) < exp(\epsilon)$$

A.3 Useful properties of DP.

DP is robust to post processing, in other words any process applied to the result of a DP algorithm is still DP. Formally:

Theorem 2. *Let $\mathcal{A}$ be (ϵ, δ)-DP algorithm and let f be any (possibly randomized) function on the range of $\mathcal{A}$, then the composition $f \circ \mathcal{A}$ is also (ϵ, δ)-DP.*

For example, in this work we used the robustness to post processing when we stated that clipping the result of the histogram was still DP, because the original histogram was DP.

Suppose we apply multiple *DP* algorithms to the dataset. Can we still say something about the privacy loss in this case? DP behaves nicely with respect to composition:

Theorem 3. *Suppose $\mathcal{A}_1 \dots \mathcal{A}_k$ are all (ϵ, δ)-DP, then an adaptive composition of them is $(k\epsilon, k\delta)$-DP.*

Adaptive in the above definition means that the algorithm $\mathcal{A}_i$ can make choices based on the outcomes of $\mathcal{A}_1, \dots \mathcal{A}_{i-1}$.

For example, in this work we used the composition theorem when took a DP-algorithm to compute the vocab and a separated DP algorithm to train the model, claiming that the entire process is DP.

There are more advanced composition theorem for DP, but these are beyond the scope of this introduction.

B A tighter bound on the differential privacy of the vocab.

In the main text, we proved a theorem about differential privacy (DP) of histograms. In the proof we bounded separately the contributions of new words by an example, and already existing words. Here we will provide stricter analysis for the case we used in this paper, by bounding both contributions together.

Given a textual dataset over the set of words X, partition the dataset into a sequence D of N-word tuples. For each tuple v, define its word histogram $f_v : X \to \mathbb{R}$ as:

$$f_v(x) = \begin{cases} 1, \ \textit{if } x \in \textit{Supp}(v) \\ 0, \ \textit{otherwise} \end{cases}$$

Let $\mathcal{D}$ denote a set of possible dataset of N-tuples. Construct $h : \mathcal{D} \to \mathbb{R}^X$ as follows:

1. Set $h'(D)(x) = \sum_{v \in D} f_v(x)$.
2. For all coordinates $x \in \textit{Supp}(h')$ add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the x coordinate in $h'(D)$.
3. Clip h' as follows to get h:

$$h(D) = \begin{cases} h'(D), \ \textit{for } h'(D) \geq C \\ 0, \ \textit{otherwise} \end{cases}$$

Then we have:

Corollary 2. *Let $\sigma > 0, 0 < \delta < 1.25e^{-3/2}$, and let $C = 1 + \sigma \operatorname{erf}^{-1}(1 - \delta/N)$. Then, the above procedure yields an (ϵ, δ)-DP histogram h, with $\epsilon = \frac{\sqrt{N}}{\sigma}\sqrt{2\log(1.25/\delta)}$.*

Proof. Let $D, D' = D \cup \{v\} \in \mathcal{D}$. We note that if there are ℓ elements $x_1, \dots x_\ell \in X$ in $\textit{Supp}(v)$ which are not in $\textit{Supp}(D)$, then when we restrict f_v to $\textit{Supp}(D)$, its norm can be bounded by $\|f_v|_{X-\{x_1 \dots x_\ell\}}(x)\|_2 \leq N - \ell$, where equality is achieved when the support of the restriction is a single element.

By (Dwork et al., 2014) We can therefore obtain $(\epsilon, \Delta_1(\ell))$-DP for the restriction to $\textit{supp}(D)$ with $\epsilon = \frac{N-\ell}{\sigma}\sqrt{2\log(1.25/\Delta_1(\ell))}$ or $\Delta_1(\ell) = 1.25 exp(-\frac{1}{2}(\frac{\epsilon\sigma}{N-\ell})^2)$.

Hence:

$$\Delta_1(\ell) = 1.25(\delta/1.25)^{(\frac{N}{N-\ell})^2}$$

For each x_i the probability to get non-zero count is smaller then δ/N. Therefore the part outside $\textit{supp}(D)$ is $(0, \Delta_2(\ell)$-DP with

$$\Delta_2(\ell) = \delta\ell/N$$

by the union bound.

Therefore, for any $0 \leq \ell \leq N$, we can bound the δ-term by:

$$\Delta(\ell) = \Delta_1(\ell) + \Delta_2(\ell) = 1.25(\delta/1.25)^{(\frac{N}{N-\ell})^2} + \delta\ell/N$$

To simplify notations we denote $y = \frac{N-\ell}{N}$, in order to prove (ϵ, δ)-DP, it is enough to show that $\Delta(y) \leq \delta$ for all $0 < y \leq 1$, we have:

$$\Delta(y) = 1.25(\delta/1.25)^{\frac{1}{x^2}} + \delta(1 - x)$$

Taking the derivative:

$$\Delta'(y) = \frac{5(\frac{\delta}{1.25})^{1/y^2} ln(1.25/\delta)}{2y^3} - \delta$$

And the second derivative:

$$\Delta''(y) = \frac{5(\frac{\delta}{1.25})^{1/y^2} ln(1.25/\delta)(3y^2 + 2ln(\delta/1.25))}{2y^6}$$

If $\delta < 1.25e^{-3/2}$, we have $\Delta''(y) > 0$ for $0 < y \leq 1$, we can also see that $\Delta(y = 1) = \delta$ and $\lim_{y \to 0^+} \Delta(y) = \delta$, therefore $\Delta(y) \leq \delta$ for $0 < y \leq 1$, and we proved (ϵ, δ)-DP.

□