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 on a plethora of language understanding tasks, 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 model, but this usually comes at the expense of model performance. Moreover, it is hard to tell given a privacy parameter ϵ what was the effect on the trained representation. In this work we aim to guide future practitioners and researchers on how to improve privacy while maintaining good model performance. We demonstrate how to train a differentially-private pre-trained language model (i.e., BERT) with a privacy guarantee of $\epsilon = 1$ and with only a small degradation in performance. We experiment on a dataset of clinical notes with a model trained on a target entity extraction task, and compare it to a similar model trained without differential privacy. Finally, we present experiments showing how to interpret the differentially-private representation and understand the information lost and maintained in this process.

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 language models. 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 amplified in medical domains, where patients data might leak and expose Protected Health Information (PHI).

One solution for pre-training the model while preserving patients' privacy is to train the model with a differential privacy guarantee. However, for a sufficiently small privacy parameter ϵ, this usually comes at the expense of model performance. Also, it was 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 (our model has 109M trainable parameters), transformer-based language models introduce an additional privacy concern, as their reliance on WordPiece based tokenization algorithm can also potentially leak private information.

Moreover, even 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 training.

In this work we provide here a detailed solution to training a differentially-private contextual embedding model, and to better understand the resulting representation. We start by presenting a method for training BERT, a contextual embedding

[*]Work was done while at Google.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 21–29

https://doi.org/10.26615/978-954-452-056-4_003

model, on medical data with a strong privacy guarantee of $\epsilon = 1$ and with only a small degradation in performance (Section 3). Possibly the most major technical challenge in doing so is the fact that the training batch size has to be fairly large, all the while training on specific hardware (TPUs) in which the batch size is limited. We overcome this obstacle by distributing each training batch over time during the training process, along with other useful manipulations (Section 2.1). As these models gain from retraining the WordPiece algorithm on the target dataset, we propose a differentially-private WordPiece algorithm, preventing additional information leakage through the model's vocabulary (Section 3.2).

After training the differentially-private BERT on clinical notes, we follow common wisdom and provide privacy tests to show that information leakage has been prevented in this process (Section 5). We further provide adversarial attacks that can help understand the privacy guarantees in terms of memorized words and phrases. These tests, when combined, provide a useful toolbox for understanding how "private" is the differentially-private model.

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., 2006b; Dwork, 2011; Dwork et al., 2014) is a framework that quantifies the privacy leaked by some randomized algorithm accessing a private dataset. In the context of training a machine learning model on private data, it enables one to bound the potential privacy leakage by releasing 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 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. In the context of language models, evaluation becomes even trickier. 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 sequence into the train corpus. The sequence itself consists of i.i.d. random WordPieces where the middle is the secret. The model is then trained on the modified corpus and evaluated for each planted sequence by trying to predict the secret WordPiece.

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. As such, attempting to train a transformer model such as BERT using the DP-SGD algorithm and without any modifications will usually lead to 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, training will not guarantee that there is no information leakage. 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 the task of entity extraction (EE) 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 Protected Health Information (PHI) as well as additional 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. Although MIMIC-III has undergone a de-identification process aimed to remove revealing information such as names and dates, the corpus and its derivative models are not considered public, and their use must adhere to certain restrictions. As a consequence, a need arises to build a medical BERT model with substantial differential privacy guarantees on its use of MIMIC-III, and this work aims to do exactly that.

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, and the 712M word MIMIC-III corpus. The training process consists of the following three steps:

(i) Build the vocab from the MIMIC-III corpus.
(ii) Train BERT from scratch on the Wikipedia + BookCorpus using the new vocab.
(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 1 and 3 without causing a significant performance loss. We discuss these problems in the following two subsections.

3.2 Constructing a differentially private vocabulary

Transformer-based models commonly tokenize inputs into WordPieces using the WordPiece algorithm. The WordPiece algorithm (Wu et al., 2016) 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 WordPieces in the training data. To prevent such leakage, we modify this algorithm to be differentially private. We do so as follows.

The WordPiece algorithm starts with constructing the word histogram of the corpus. This histogram is then manipulated to obtain the WordPiece output vocabulary. Since differential privacy is robust to post-processing, it is enough to make the input histogram differentially private in order to guarantee a differentially-private end-result vocabulary. Our differentially-private WordPiece algorithm is therefore to add noise to the histogram with given privacy parameters and apply the standard WordPiece algorithm.

Histogram noising is done following (Korolova et al., 2009; Bun et al., 2019), let X be the set of all possible n distinct words. For the input histogram $h : X \to \mathbb{R}$, we do:

(i) For all $x \in X$, if $h(x) > 0$, add Laplace noise: $h(x) \leftarrow h(x) + \text{Lap}(2/\epsilon)$.

(ii) For all $x \in X$, if $h(x) < 1 + 2\ln(2/\delta)/\epsilon$, set $h(x) \leftarrow 0$.

The output h of this process satisfies (ϵ, δ)-differential privacy with respect to replacing one of the words in the histogram counts. Assuming $0 < \epsilon < ln(n)$, $0 < \delta < 1/n$ (Bun et al., 2019; Korolova et al., 2009).

In order to obtain differential privacy at the level of BERT example (256???? WordPiece) we use the basic composition theorem for non-adaptive queries (Dwork et al., 2006a; Dwork and Lei, 2009):

Theorem 1 *Let $M_1, \ldots M_k$ be (ϵ, δ)-differentially private, then $(M_1, \ldots, M_k)$ is $(k\epsilon, k\delta)$-differentially private.*

We used parameters $\epsilon' =?, \delta' =??$ in the noisy histogram algorithm above to achieve an example level ($\epsilon = 256*?, \delta = 256*?$) differential privacy.

3.3 Training a differentially private BERT

We use the DP-SGD method supplied 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 trade-off privacy against performance. Moreover, 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 code. 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 the sharded batching. Let B denote the unsharded

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

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 throughout this project, we needed larger batch sizes. 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, we couldn't get beyond parallelism of $P = 256$ with sharded batch size of 32, achieving total batch size $B = 8192$.

The way we chose to solve this problem is to spread the batch in time, so `apply_gradients` is called only after T batches are processed with the average total 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 improve performance while preserving privacy. We focus on the medical domain as it has strict privacy requirements and its language is distinct enough such 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 the training scheme described in Section 3. At this point, we use a WordPiece vocabulary generated from MIMIC-III without privacy guarantees.

Entity-extraction task. For the supervised task training, we use i2b2-2010, a dataset from the i2b2 National Center for Biomedical Computing for the NLP Shared Tasks Challenges (Uzuner et al., 2011). This dataset contains clinical notes tagged for concepts, assertions, and relations. In this task, 170 patient reports are labeled with three concepts: test, treatment, and problem. The total number of entities in each category are as follows:

- Problem: 7,073
- Test: 4,608
- Treatment: 4,844

We perform 5-fold cross validation where each fold has random training (136 notes) and test (34 notes) sets.

Baselines. We compare our differentially private BERT model, denoted as DP BERT, 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 aforementioned EE task. 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.

5.1 Preserving Useful Information

For our first experiment, we pre-trained a DP BERT model, then evaluated on an EE task over the i2b2-2010 dataset. We summarize our results in the following table.

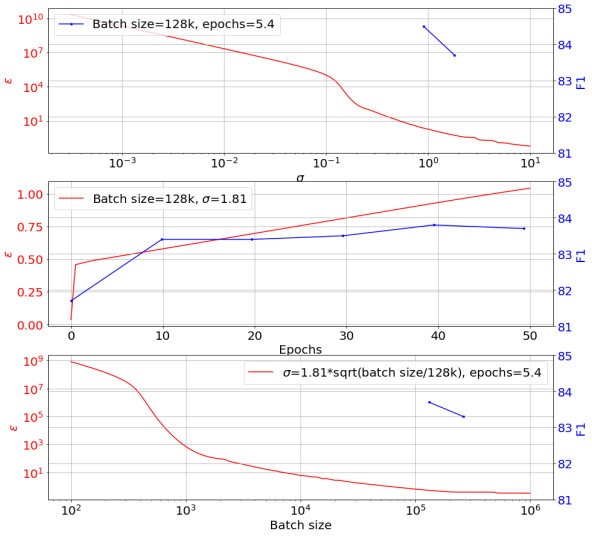

Figure 1: Top to bottom - privacy parameter ϵ (red) and test F1 score on the EE task (blue), as a function of: noise multiplier σ; number of pre-training epochs; pre-training batch size.

Model	ϵ	F1 Score
BERT	∞	76.3%
BERT-M	∞	86.8%
BioBERT	∞	86.5%
BERT-M	3.2	84.5%
BERT-M	1	83.7%

Table 1: Results on the Medical Entity Extraction task. $\epsilon = \infty$ means no differential privacy.

These were all evaluated after 1M training steps with batch size 128K. As one can observe, the additional pre-training either on MIMIC-III or on PubMed gives a significant boost in performance over the off-the-shelf BERT. The addition of differential privacy then deteriorates performance only slightly, and, as expected, performance is inversely proportional to ϵ (recall that smaller ϵ implies better privacy).

In addition, in Fig. 1 we evaluate the change in ϵ and of the F1 score of the downstream task as a function of batch size, noise multiplier σ, and the number of pre-training epochs. The behavior in all three parameters is as expected. Increasing σ enables more privacy (lower ϵ), but worsens performance. Similarly, with more pre-training epochs the model gathers more information about the training data, so we obtain better F1 score but worse privacy preservation (higher ϵ). When increasing the batch size, we also increase the noise multiplier

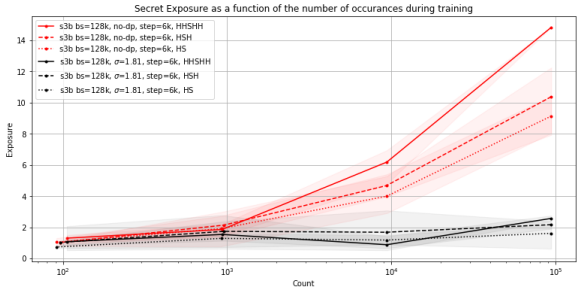

Figure 2: Secret exposure as a function of the number of secret occurrences. Black lines for models with differential privacy $\epsilon = 0.58$, red lines for models without DP $\epsilon = \infty$.

σ proportionally, thus both ϵ and the F1 decrease.

5.2 Forgetting Private Information

For our second experiment, we followed Carlini et al. (2019) to test the model's ability to memorize private information. We injected the MIMIC-III dataset with "secrets", of the form HS, HSH, and HHSHH, where H is a generic word and S is a secret word. The injection was done by sampling locations to plant each secret uniformly at random from the dataset. We tested all three forms of secrets on a DP model and a non-DP model, with different numbers of appearances of the secret in the dataset. For each such evaluation, we measured the exposure of the secret which essentially measures how well the model memorized the secret (see Carlini et al., 2019 for exact definition of "exposure"). As one can see from Fig. 2, even when the secret appears as much as 100K times in the data, the DP model performs significantly better than without differential privacy. This seems to suggest that the model learns through information that helps it generalize rather than memorizing 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 pipeline 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. To overcome the difficulties associated with learning such models, we have offered practical measures for circumventing them, most notably through vastly increasing batch sizes. Then, 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 yet measured the model's performance with the 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, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006a. Our data, ourselves: privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503. Springer.

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. Association for Computing Machinery.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006b. 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. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE.

Ilya Mironov, Kunal Talwar, and Li Zhang. 2019. Renyi 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.

Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, and J. Klingner. 2016. 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*.