A Study on Extracting Named Entities from Fine-tuned vs. Differentially Private Fine-tuned BERT Models

Andor Diera, Nicolas Lell
Ulm University
Germany
firstname.lastname@uni-ulm.de

Aygul Garifullina
BT, Ipswich
UK
aygul.garifullina@bt.com

Ansgar Scherp
Ulm University
Germany
ansgar.scherp@uni-ulm.de

ABSTRACT

Privacy preserving deep learning is an emerging field in machine learning that aims to mitigate the privacy risks in the use of deep neural networks. One such risk is training data extraction from language models that have been trained on datasets, which contain personal and privacy sensitive information. In our study, we investigate the extent of named entity memorization in fine-tuned BERT models. We use single-label text classification as representative downstream task and employ three different fine-tuning setups in our experiments, including one with Differentially Privacy (DP). We create a large number of text samples from the fine-tuned BERT models utilizing a custom sequential sampling strategy with two prompting strategies. We search in these samples for named entities and check if they are also present in the fine-tuning datasets. We experiment with two benchmark datasets in the domains of emails and blogs. We show that the application of DP has a detrimental effect on the text generation capabilities of BERT. Furthermore, we show that a fine-tuned BERT does not generate more named entities specific to the fine-tuning dataset than a BERT model that is pre-trained only. This suggests that BERT is unlikely to emit personal or privacy sensitive named entities. Overall, our results are important to understand to what extent BERT-based services are prone to training data extraction attacks.

KEYWORDS

privacy preserving deep learning, language models, training data extraction, named entities, text generation, differential privacy

1 INTRODUCTION

Deep Neural Networks (DNNs) became the *de facto* tool for achieving state-of-the-art performance in many research domains, such as computer vision and natural language processing (NLP) [35]. Although utilizing large volumes of training data is one of the main driving factors behind the great performance of DNNs, publishing these models to the public raises some serious privacy concerns regarding private and confidential information present in the training data [39]. These privacy concerns are especially relevant for large Language Models (LMs) which form the basis of state-of-the-art technologies in many NLP tasks [7, 8]. LMs are defined as statistical models which assign a probability to a sequence of words. Recent versions of these models are usually first pre-trained in a task-agnostic self-supervised manner. The latest large LMs use a corpus size ranging from hundreds of gigabytes to several terabytes of text [6, 55] during this self-supervised process. The sheer size of these datasets makes it near impossible for researchers to remove all confidential information which may be present in the corpus.

A recent study has shown that it is possible to extract personal information from some large LMs, even if that given information has only appeared once in the training corpus [8]. While the training cost of these large LMs became so prohibitively expensive that only the biggest tech companies can afford it [61], pre-trained LMs are commonly used in businesses that work with huge amounts of text data. These businesses include banks, telecommunications, and insurance companies, which often handle a great amount of personal and privacy sensitive data. In practice, pre-trained LMs are fine-tuned on a business-specific dataset using some downstream task (such as text-classification, question-answering, or natural language inference) before deployment [12]. Although the fine-tuning may mitigate some of the unintended memorization of the original dataset used in pre-training, it raises new concerns regarding the personal and privacy sensitive information in the business-specific dataset used for the fine-tuning process [8]. Privacy Preserving Deep Learning (PPDL) is a common term used for methods aiming to mitigate general privacy concerns present in the use of DNNs. Multiple approaches have been proposed to achieve PPDL [48], but there is no perfect solution to this problem, with each method having its own challenges and limitations. The most popular techniques include Federated Learning [46], the application of Differential Privacy (DP) [1, 81], encryption [4, 26], and data anonymization [63, 65].

We investigate whether it is possible to extract personal information from one of the most popular modern LMs, BERT [12]. BERT is an auto-encoder transformer that is mostly used for natural language understanding tasks. Since BERT is less adept at generating long, coherent text sequences [70], we focus our study on the generation and extraction of named entities. We conduct our experiments on three typical fine-tuning setups to understand the privacy risks involved in using BERT for commercial purposes. We consider fine-tuning all layers of BERT (*Full*), fine-tuning only the last encoding layer and the classifier head of BERT (*Partial*), and partial fine-tuning but with a privacy preserving optimizer (*Differentially Private*, short: *DP*). As a privacy preserving optimizer, we employ the established differentially private stochastic gradient descent (DPSGD) algorithm [1], which we discuss in detail in Section 3.1. We compare the fine-tuning setups with an "only pre-trained" BERT base model. We experiment with two benchmark datasets in the domains of emails (Enron Email corpus [33]) and blogs (Blog Authorship Corpus [60]). For triggering entity extraction, we use two prompting techniques. The *naive prompting* is based on randomly selected text from the web, while the *informed prompting* uses actual text from the datasets' test sets. Each experimental setup is assessed with regard to its performance on the down-stream task, i. e., single-label text classification, and the extent of named entity memorization. In summary, the insights from our experiments are:

- The memorization rate of named entities in the fine-tuned BERT models is less than 10% in both datasets across all setups. Interestingly, the fine-tuned models do not emit more entities from the fine-tuning datasets than an only pre-trained BERT model.

- When comparing the informed prompting versus the naive prompting, the BERT models consistently generate more named entities when using naive prompts. Thus a potential attacker does not require prior knowledge about the training dataset of a model.
- Applying differentially private (DP) fine-tuning results in a strong drop in the amount of memorized entities at the cost of downstream task performance. It effectively reduces the amount of entity memorization in fine-tuned BERT models.

The subsequent section discusses the related work. It serves as foundation for defining our framework and methods for extracting named entities from fine-tuned BERT models, as described in Section 3. Section 4 describes the datasets and the implementation details of the experiments. The results are described in Section 5 and discussed in Section 6, before we conclude.

2 RELATED WORK

In this section, we briefly review the literature on LMs with a focus on the BERT model and text generation, possible privacy attacks against DNNs, and existing approaches to Privacy Preserving Deep Learning (PPDL).

2.1 Language Models and Text Generation

Modern large LMs rely on two core concepts that led to their dominance in the NLP field: the focus on the self-attention mechanism in the DNN architecture and the introduction of large-scale task-agnostic pre-training to learn general language representations [73]. Self-attention is used for modeling dependencies between different parts of a sequence. A landmark study in 2017 [68] has shown that self-attention was the single most important part of the state-of-the-art NLP models of that time. It introduced a new family of models called transformers, which rely solely on stacked layers of self-attention and feed-forward layers. Besides the state-of-the-art performances, another great advantage of the transformer architecture is that unlike a recurrent architecture, it allows for training parallelization.

The ability to parallelize training, alongside the significant increase in computational power allowed these models to train on larger datasets than once was possible. Since supervised training requires labeled data, self-supervised pre-training with supervised fine-tuning became the standard approach when using these models [45]. The first transformer that achieved great success on a large-range of NLP tasks using this approach was the Generative Pre-trained Transformer (GPT) [54].

State-of-the-art transformers can be divided into three main categories based on their pre-training approach [78]. Auto-regressive models use the classical language modeling pre-training task of next word prediction. Auto-encoding models are pre-trained by reconstructing sequences that have been corrupted in some way. Sequence-to-sequence models usually employ objectives of encoding-decoding models for pre-training, like replacing random sequences in a text with one special token with the objective of predicting that given sequence [55].

2.1.1 *BERT.*

A major limitation of the auto-regressive models is that during pre-training they learn a unidirectional language model. In these models, tokens are restricted to only attend other tokens left to them, therefore they can only predict a token based on the context from the left. In contrast, BERT is an encoder-only model that uses an attention mechanism on the entire input sequence [12]. This model utilizes Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) as pre-training tasks. In MLM, some tokens are randomly removed from the input sequence and the model is trained to predict the removed tokens using context from both directions. Formally, the objective is to learn the probability distribution over vocabulary V for masked token x_{mask} in sentence X such that

$$\mathrm{x}_{mask} = \arg\max_{w \in \mathcal{V}} \Pr(\mathrm{w}|\mathrm{X} \backslash \mathrm{x}_{mask})$$

Although the parameter count of BERT is greatly surpassed by more recent large LMs (such as GPT-3 [6]), BERT is still one of the most common baselines in many NLP benchmark tasks. The strong performance coupled with the fact that the model is democratized and has publicly available pre-trained implementations makes BERT a popular choice of NLP model both in industry and academia. Since its original release, there have been dozens of follow-up studies and models published [57]. The most notable variants include RoBERTa an improved version of BERT which utilizes advanced hyperparameter optimization and a modified MLM method on a larger pre-training corpus [41], ALBERT a “lite” version of BERT using factorized embedding parameterization, cross-layer parameter sharing, and sentence-order prediction instead of NSP during pre-training [34], and DistilBERT, a general purpose distilled model of the pre-trained BERT [59]. Some variants of the BERT model employ domain specific pre-training corpora for increased performance on domain specific tasks: SciBERT [5] uses scientific texts, ClinicalBERT [3] is pre-trained on a clinical text corpus, while finBERT [42] uses a large financial corpus.

2.1.2 *Natural Language Generation.*

Natural Language Generation (NLG) is a subfield of NLP that is focused on producing natural language text that enables computers to write like humans [78]. It lies at the core of many applied NLP domains such as machine translation [79], text summarization [50], and dialogue systems [9]. Early approaches in text generation used methods like recurrent neural networks [17], deep generative models [32], and variational autoencoders [32], all to varying degree of success. The introduction of large pre-trained transformer strongly improved the state-of-the-art in NLG.

Although auto-regressive transformer models are the standard choice for the task of NLG (since they are already trained to predict the next token based solely on previous tokens in a sequence), it has been shown that BERT can also be utilized to generate reasonably coherent text. Wang and Cho [70] designed a generation strategy for BERT based on Gibbs sampling [24], where given a seed sequence, tokens at random positions are masked and replaced by new tokens based on the sampling technique. Another generation strategy developed for auto-encoding transformers [25] is to use a fully masked sequence as input and predict all tokens at once. Subsequently, tokens with the lowest probability are iteratively re-masked and replaced with a newly computed token.

2.2 Privacy Attacks in Machine Learning

Privacy attacks in machine learning (ML) denote a specific type of adversarial attack, which aim to extract information from a ML model. Based on recent surveys in the field [11, 39, 56], these attacks can be divided into five main categories.

2.2.1 *Membership Inference Attacks.*

The goal of a membership inference attack is to determine whether or not an individual data instance is part of the training dataset for a given model.

This attack typically assumes a black-box query access to the model. The common approach to this type of attack is to use a shadow training technique to imitate the behavior of a specific target model. In shadow training, a ML model (shadow model) is trained on a dataset that has a disjoint but identically formatted training data as the target model. The trained inference model is then used to recognize differences on the target model predictions between inputs used for training and inputs not present in the training data [62]. Membership inference attacks have been shown to work on models used for supervised classification tasks [62], Generative Adversarial Networks (GANs) [40], variational autoencoders [40], and the embedding layers of LMs [44].

2.2.2 Model Extraction Attacks. The adversarial aim of a model extraction attack is to duplicate (i. e., "steal") a given ML model. It achieves this by training a function f' that is approximating the function f of the target model [39]. A shadow training scheme has been shown to successfully extract popular ML models such as logistic regression, decision trees, and neural networks, using only black-box query access [67]. Other works have proposed methods to extract information about hyperparameters [71] and properties of the architecture [51] in neural networks.

2.2.3 Model Inversion Attacks. The idea behind model inversion attacks is that an adversary can infer sensitive information about the input data using a target model's output. These attacks can be used to extract input features and/or reconstruct prototypes of a class in case the inferred feature characterize an entire class [11]. The first model inversion attack [20] was based on the assumption that the adversary has white-box access to a linear regression model, with some prior knowledge about the features of the training data. With the use of the output labels and known values of non-sensitive features, this attack is capable of estimating values of a sensitive feature. Existing work was later extended to neural networks with a new type of model inversion attack [19], which reformulated the attack as an optimization problem where the objective function is based on the target model's output and uses gradient descent in the input space to recover the input data point. This technique allows the adversary to reconstruct class prototypes (i. e., faces in a facial recognition model) given a white-box access to the model and knowledge about the target labels with some auxiliary information of the training data.

2.2.4 Property Inference Attacks. The goal of property inference attacks is to infer some hidden property of a training dataset that the owner of the target model does not intend to share (such as feature distribution or training bias). Initially, property inference attacks were applied on discriminative models with white-box access [47, 53]. A more recent work has extended the method to work on generative models with black-box access [53].

2.2.5 Training Data Extraction Attacks. Training data extraction attacks aim to reconstruct training datapoints, but unlike model inversion attacks, the goal is to retrieve verbatim training examples and not just "fuzzy" class representatives [8]. These attacks are best suited for generative sequence models such as LMs. Initially these attacks have been designed for small LMs using academic datasets [7, 66, 77]. The aim of these studies was to measure the presence of specific training datapoints in the text samples generated by the models. A common approach to measure the extent of this unintended memorization is to insert so-called "canaries" (artificial datapoints) into the training datasets and quantify their occurrence during sequence completion [7]. Since these initial studies were based on smaller models trained with a high number of epochs, it was assumed that this kind of privacy leakage must be correlated with overfitting [77]. However, a follow-up study using the GPT-2 model, which is trained on a very large corpus for only a few epochs, showed that even state-of-the-art large LMs are susceptible to these kinds of attacks. Using the pre-trained GPT-2 model, Carlini et al. [8] were able to generate and select sequence samples which contained low k-*eidetic* data-points (data points that occur k times in the training corpus). A study by Lehman et al. [37] on Clinical BERT attempted to extract patient-condition association using both domain specific template infilling and the text generation methods inspired by the text extraction research done on GPT-2 [8] and the BERT specific text generation technique proposed by Wang and Cho [70]. Their methods were not successful in reliably extracting privacy sensitive information (patient-condition associations) from Clinical BERT, but it remains inconclusive whether its due to the limitations in their method or in the linguistic capabilities of BERT.

2.3 Privacy Preserving Deep Learning

Since deep learning is a subfield of ML, most methods developed for privacy preserving ML can be also adapted to Privacy Preserving Deep Learning (PPDL). Based on the literature [11, 39, 56], these methods can be divided into five main categories.

2.3.1 Encryption. Cryptography-based methods can be divided into two subcategories, depending whether the target of the encryption is the training data [26] or the model [4]. Regardless of the target, most existing approaches use homomorphic encryption, which is a special kind of encryption scheme that allows computations to be performed on encrypted data without decrypting it in advance [2]. Since training a DNN is already computationally expensive, adding homomorphic encryption to the process raises major challenges as it increases training times by at least an order of magnitude [39].

2.3.2 Data Anonymization. Data Anonymization techniques aim to remove all Personally Identifiable Information (PII) from a dataset. The common approach to achieve this is to remove attributes that are identifiers and mask quasi-identifier attributes [74]. The popular k-anonymity algorithm [65] works by suppressing identifiers (i. e., replacing them with an asterisk) and generalizing quasi-identifiers with a broader category which has a frequency of at least k in the dataset. Although data anonymization techniques were developed for structured data, it is possible to adapt them to unstructured text data [27] as well as jointly anonymizing structured data and unstructured text data [63].

2.3.3 Differentially Private Learning. Differential Privacy (DP) is a rigorous mathematical definition of privacy in the context of statistical and machine learning analysis. It addresses the challenge of "learning nothing about an individual while learning useful information about the population" [16]. In ML, DP algorithms aim to obfuscate either the training data [80] or the model [58] by adding noise. Since directly adding noise to DNN parameters may significantly harm its utility, the best and most common place for applying DP in deep learning is the gradients [82]. Abadi et al. [1] proposed an efficient training algorithm with a modest privacy budget called Differentially Private Stochastic Gradient Descent (DPSGD). DPSGD ensures DP by cutting the gradients to a maximum L2 norm for each layer and then adding noise to the gradients. Although DPSGD comes with increased computational cost and performance loss, variations of this algorithm [10, 14] still belong to the cutting-edge of PPDL research.

2.3.4 Aggregation. Aggregation methods are generally used along with distributed/collaborated learning, in which multiple parties join a ML task while aiming to keep their respective datasets private [39]. The most popular collaborative framework for privacy preservation is Federated Learning introduced by McMahan et al. [46]. Although aggregation methods can provide data security during distributed training, their privacy preserving aspects are more limited than other PPDL approaches.

2.3.5 Combined Approaches. The four main categories of PPDL methods are not mutually exclusive. DP is often used in collaborated learning where it is combined with aggregation techniques [72]. A promising framework called Private Aggregation of Teacher Ensembles (PATE) [52] proposes improved privacy preservation with the use of an ensemble of teacher models (which have been trained on non-overlapping datasets), and a differentially private aggregation mechanism. The knowledge of the aggregated model is then transferred into a student model, resulting in a model with strong privacy guarantees.

2.4 Summary

Transformer-based architectures achieve the state-of-the-art in most areas of NLP. The BERT model [12] is widely applied in most NLP tasks. It is not pre-dominantly used for text generation due to its bidirectional pre-training approach, but can it can be utilized for auto-regressive text generation like for sentences [70] or labels [31]. From such generated text output, it is possible to extract various type of information from ML models using privacy attacks [11, 39, 56]. Large LMs are prone to training data extraction attacks. For example, a study on GPT-2 has shown that it is possible to extract privacy sensitive information from the training data with black-box access, using different prompting techniques when generating text [7]. As privacy attacks become more frequent against DNNs, the role of privacy preservation in deep learning also becomes more crucial. Although there are multiple ways to improve the privacy of a model including the use Differential Privacy during training [1], there is no perfect solution as of yet [11, 39, 56].

3 EXTRACTING MEMORIZED NAMED ENTITIES FROM BERT

In order to extract the named entities of the fine-tuning dataset from the BERT model, we present the experimental pipeline depicted in Figure 1. The pipeline consists of three main phases: the fine-tuning (including a privacy preserving approach using Differential Privacy), the text generation from the fine-tuned models, and the evaluation of the named entity memorization.

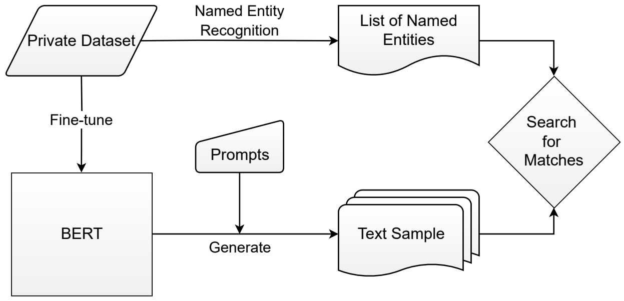

Figure 1: An illustration of our framework for extracting training data entities from BERT. First, we fine-tune a pre-trained BERT on a private dataset. Next, we generate text samples from the fine-tuned model using prompts. Finally, we search the generated samples for the named entities that occur in the private dataset.

3.1 Fine-tuning

In the fine-tuning phase, we employ single-label text-classification as the downstream task. Our setup consists of three different fine-tuning methods: *Full*, *Partial*, and *Differentially Private (DP)* fine-tuning. The different fine-tuning methods are depicted in Figure 2.

The *Full* setup follows the standard practices of fine-tuning LMs, where a classifier head is attached to the base network and all the weights of a pre-trained network along with the classifier head are retrained on the task-specific dataset with a low learning rate [30]. Full fine-tuning usually leads to the best results on the downstream task, but in the case of large LMs, it is not always feasible due to the size of these networks and the computational costs of retraining them. Due to this constraint, researchers have designed alternative fine-tuning strategies where fine-tuning is employed in a more optimized manner [30, 36]. A common alternative strategy is to freeze most of the layers in a network and only retrain the last few encoder layers with task-specific head of the network [43, 64]. In our *Partial* setup, we freeze all layers of the BERT model except for the last encoding layer. Applying DP in fine-tuning puts additional noise to the gradient updates, which in the lower layers carries a detrimental effect to the pre-trained knowledge of the model as the weights of the bottom layers are more sensitive to noise. For this reason, in the *Differentially Private* setup we employ the same layer-freezing approach as in the *Partial* setup.

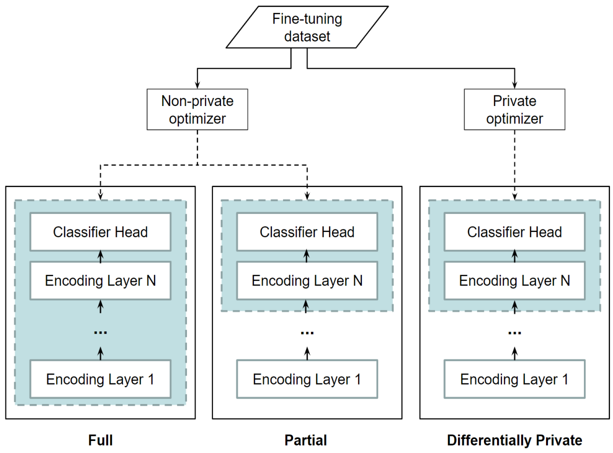

Figure 2: An illustration of different fine-tuning methods. In the *Full* setup (left), we apply standard fine-tuning to all layers. In the *Partial* setup (middle), we only fine-tune the last encoding layer and the classifier head. For the *Differentially Private* setup (right), we also employ partial fine-tuning but use the privacy preserving optimizer DPSGD [1].

In addition, the DP fine-tuning method uses the Adam variant of DPSGD by Abadi et al. [1]. DPSGD is a modification to the stochastic gradient descent algorithm that employs (ϵ, δ) differential privacy [15]. In the formal definition of (ϵ, δ) differential privacy, a randomized algorithm M is differentially private if:

$$\Pr[M(x) \in S] \leq \exp(\epsilon) \cdot \Pr[M(y) \in S] + \delta,$$

where S denotes all the potential output of M that can be predicted, ϵ is the metric of privacy loss (also known as the privacy budget) at a differential change in the data (e. g., adding or removing one datapoint), and δ is the probability of an accidental privacy leak. In deep learning, an ϵ value is defined as modest when it is below 10 and δ is usually set to the the reciprocal of the number of training samples [14, 76]. In standard data analytics settings, ϵ values between 0 and 1 are considered to be highly private, and values between 2 and 10 are considered somewhat private. However, in deep learning it is hard to achieve a ϵ value under 1 due to continuous increase of ϵ between training epochs.

The DPSGD algorithm entails two major changes to the gradient descent algorithm: the introduction of gradient clipping and the addition of Gaussian noise to the clipped gradients. The clipping limits how much an individual training point can impact the model parameters, while the addition of the noise randomizes the behavior of the algorithm, making it statistically impossible to know whether or not a training point was included in the training set. These modifications to the gradients happen on a microbatch level (ideally for each individual training sample) and are aggregated for the standard batch optimization step.

3.2 Text Generation

Although the standard objective of NLG is to produce text that appears indistinguishable from human-written text (see related work in Section 2.1.2), in our study we are less interested in general text quality in terms of coherence or grammatical correctness. Our primary goal is to trigger the fine-tuned models to generate named entities found in the training data. To achieve this, we employ two different prompting methods and an efficient generation strategy that produces diverse text samples.

3.2.1 Prompt Selection. Selecting good prompts is a crucial step in triggering the model to unveil information about the training data. We employ two different prompting strategies for text generation.

(1) In the first one, we take the strategy shown to achieve the best results in the experiments of Carlini et al. [8], in which a fixed length of string sequence is randomly sampled as prompt from the Crommon Crawl[1] dataset. This we refer to as a *naive prompting*, since we randomly use text samples scraped from the internet. The selected prompts likely have no or only very little connection with the text and named entities from the fine-tuning dataset.
(2) In the second strategy, we create the prompts by randomly selecting string sequences of a fixed length from the test set of the fine-tuning data. This setup is considered as *informed prompting* given that the prompts come from the same domain and are generally highly similar to the training data.

3.2.2 Text Generation. Despite the fact that the bidirectional nature of BERT does not naturally admit to sequential sampling, Wang and Cho [70] have shown that it is also possible to utilize this strategy for BERT. Although their results suggest that their non-sequential iterative method produces slightly more coherent text than sequential sampling, it requires multiple iterations for each token. Since text coherence is not our primary goal, we choose to employ a computationally less expensive sequential sampling method. In this method, we choose a randomly selected prompt (see above) as a seed sequence and extend it with a masked token. For each iteration, we predict the masked token and replace the mask. We add an additional masked token to the extended sequence until we reach the defined sequence length.

On top of this generation method, we also employ a combination of beam search and nucleus sampling as an additional decoding strategy. Beam search is a commonly used decoding method in machine translation tasks [21]. Compared to greedy search, where at each iteration only the token with the highest probability is selected, beam search selects multiple tokens at each iteration. The number of tokens is defined by the beam width parameter and an additional conditional probability is used to construct the best combination of these tokens in a sequence. While both greedy search and beam search select tokens based on maximum likelihood, sampling from the probability distribution is also a viable method. The most popular sampling method is top-k sampling, where in each iteration a token is sampled from a set of k candidates with the highest probability. Nucleus sampling (also called top-p sampling) is an alternative strategy to top-k, where instead of using a set with a fixed length, the smallest possible set is constructed with tokens whose cumulative probability exceeds the probability value of parameter p [28]. These sampling methods can be combined with both search algorithms.

3.3 Evaluating Named Entity Memorization

Named entities refer to objects and instances that identify one item from a set of other items sharing similar attributes. They usually include entity types like (person) names, organizations, locations, products, and special temporal or numerical expressions like dates or amounts of money [49]. In order to evaluate the extent to which the models have memorized the named entities found in the fine-tuning dataset, we first extract them from the datasets using Named Entity Recognition (NER) and create a dictionary with the entities and their corresponding entity types.

After this, we create three different entity sets from this dictionary. The first one (*All*) consists of every entity with a character length greater than 3. In the second (*Private*), we do a cross-check of the first entity set with the pre-training data, and remove all entities also present in the pre-training datasets, leaving us a set of entities that only appear in the fine-tuning data. For the third and final set (*Private 1-eidetic*), we keep all 1-eidetic entities, i. e., entities which appeared only once in the fine-tuning dataset, of the second set and discard everything else. Once we have these three sets and the generated samples from each model setup (including the samples generated from a base model that was not fine-tuned), we count the number of exact matches in the samples.

4 EXPERIMENTAL APPARATUS

4.1 Datasets

For selecting suitable datasets for the study we had two criteria. First, to avoid privacy issues and ethical concerns only publicly available datasets were chosen. Second, to provide a good basis for the measurement of memorization, we were interested in datasets that contain a large number of named entities. We choose data which has English as its primary language and kept 20% of each dataset for testing. The main characteristics of the datasets can

[1] http://commoncrawl.org

be seen in Table 1. The label distribution of each dataset is shown in Appendix A.

Table 1: Characteristics of the datasets

Dataset	N	#Train	#Test	#Classes
Enron	7,501	6,000	1,501	7
BlogAuthorship	430,269	344,215	86,054	39

4.1.1 Enron Email Dataset. The raw Enron Email corpus [33] consists of 619,446 email messages from 158 employees of the Enron corporation, made public due to legal investigations. The cleaned version has 200,399 messages and is commonly used for various NLP tasks. Since this dataset contains full emails of real users, it includes naturally occurring personal information such as names, addresses, organizations, social security numbers etc. This high amount of entities makes the dataset a perfect fit for our experiment.

Since the original dataset is not fitted to text-classification (as it lacks any official labels), we adapt it by labeling the emails by the folders' names they are attached to (i. e., "sent-mail", "corporate" , "junk", "proposals" etc.). In total there are 1,781 folders in the dataset but most of the folders contain less than a couple of emails. Therefore, we first selected the folders with more than a 100 emails, and hand-picked folders that could be considered as valid classes in an applied setting. This leaves us with seven classes, which we employ for the downstream task of single-label text classification. The topics of these seven classes are "logistics", "personal", "management", "deal discrepancies", "resumes", "online trading", and "corporate". During the preprocessing of the emails, we removed the forward blocks, HTML links, line breaks, and tabs. The removal of forward blocks and HTML links was especially important to improve the quality of the generated texts. A more detailed description of the preprocessing can be found in Appendix A.

4.1.2 Blog Authorship Corpus. The Blog Authorship Corpus [60] contains text from blogs written until 2004, with each blog being the work of a single user. The data was collected from blogger.com in 2004. The corpus incorporates a total of 681,288 posts from 19,320 users. Alongside the blogposts, the dataset includes topic labels and demographic information about the writer, including gender, age, and zodiac sign. Although the blogposts were written for the public, they contain some PII such as names, organizations and postal addresses. We adapt the dataset for text-classification by labeling the posts by the blogs' topics.

Posts with the topic label "unknown" were removed. After the removal, the dataset consists of 430,269 posts with 39 unique labels. Preprocessing on the blogposts was kept to a minimal: non-printable ASCII characters, non-ASCII characters (e. g., Korean letters), and URL links were removed, otherwise the text remained unchanged.

4.2 Procedure & Implementation

The procedure of our experiments follows the pipeline as illustrated in Figure 1. Below, we describe the details of each step. All experiments were conducted with the same random seed on a NVIDIA A100 HGX GPU with 40 GB of RAM.

4.2.1 Fine-tuning. The experiments are based on the Hugging Face implementation of the BERT base uncased model [73], which uses 12 layers of transformer blocks and is pre-trained on the English Wikipedia [18] and Book Corpus [83] datasets. For single-label classification, a custom classifier head is attached to the base model consisting of a Dropout and a Linear layer. In the *Full* and *Partial* setup we used the standard Adam optimizer, while in the *Differentially Private* fine-tuning, we changed it to the DPAdam optimizer from the Opacus library [75] with a microbatch size of 1.

4.2.2 Text Generation. For text generation, we first removed the classifier heads from the fine-tuned models and attached a pre-trained MLM head instead. We then used the sequential generation method described in Section 3, with the addition of beam search and nucleus sampling combined with a temperature parameter. During prompt creation, we sampled a 100 character length string either from the Common Crawl dataset (naive prompting) or from the test set (informed prompting). We set the sequence length to 256 tokens. We removed the tokens of the prompt before saving the samples, i. e., if the prompt contained any entities, they are not considered for the evaluation. In total, we generated 20,000 text samples for each setup.

4.2.3 Named Entity Recognition. For collecting the named entities from the fine-tuning datasets, we employed the NER system of the spaCy library that utilizes a custom word embedding strategy, a transformer, and a transition-based approach to named entity parsing [29]. spaCy distinguishes between a total of 18 different named entity types. Out of these 18 entity types we selected seven (Person, Organization, Location, Geo-Political Event, Facility, Money, Cardinal), which have a high possibility to contain personal or privacy sensitive information. A more detailed discussion about the entity type selection is found in Appendix B. When creating the *Private* entity set described in Section 3.3, we cross-checked our fine-tuning entities with the pre-training datasets (the Book Corpus and Wikipedia datasets, available through the datasets library [38]) to discard the entities present both in fine-tuning and pre-training. The numbers of named entities per type in each of the three sets can be seen in Table 2.

4.3 Hyperparameter Optimization

4.3.1 Fine-tuning. During fine-tuning, we carefully optimized the models on both datasets using manual tuning based on test accuracy. Dropout rates were fixed based on the default BERT base implementation of the huggingface library (0.1 for attention dropout and 0.3 for the classifier) [73]. For batch size, learning rate, and number of epochs a search space was defined based on previous works [13, 22]. Specifically, we chose the batch size from {8, 16, 32}, the learning rate from {5e-3, 1e-3, 1e-4, 5e-5, 1e-5}, and the number of epoch from {3, 5, 10}. Across all setups we found that using a batch size of 32 leads to the best performance. On the Enron dataset the highest accuracy values were achieved when the number of epochs is set 10, while the Blog Authorship dataset required 5 epochs to reach the highest values. In the *Full* setup a learning of 1e-5 was found to be the best performing on both datasets. In the *Partial* setup the best results were found with a 5e-5 learning rate for the Enron dataset and 1e-4 for the Blog Authorship dataset.

In the *DP* fine-tuning, the best results were achieved with a learning rate of 1e-3 on both datasets. In this setup two additional hyperparameters had to be optimized to achieve the highest

Table 2: Number of named entities found in the datasets sorted by type. *All* refers to the entity set which contains every entity found in the fine-tuning dataset. *Private* refers to the set of entities *only found* in the fine-tuning datasets, i. e., we removed all entities also present in the pre-training datasets. Finally, *Private 1-eidetic* contains only the 1-eidetic entities from *Private* set. See Section 3.3 for details on the definition and creation of the three entity sets *All, Private,* and *Private 1-eidetic.*

Named Entity Type	Enron			Blog Authorship		
	All	Private	Private 1-eidetic	All	Private	Private 1-eidetic
PERSON	10,712	7,717	4,844	209,434	137,892	113,599
ORG	9,933	7,178	5,001	168,068	107,480	90,594
LOC	316	175	125	10,562	4,902	4,049
GPE	1,551	739	490	37,691	17,781	13,196
FAC	367	230	174	12,824	7,137	6,349
MONEY	1,220	736	585	11,216	7,551	6,343
CARDINAL	2,918	1,924	1,386	24,075	13,020	10,810
Total	27,017	18,726	12,605	473,870	295,763	244,940

possible accuracy while keeping the privacy budget ϵ single-digit. We set the per-example gradient clipping threshold to 10 based on a previous study using DP with BERT [76], and found the best values for the noise multiplier to be 0.5 for the Enron and 0.4 for the Blog Authorship dataset.

4.3.2 Text Generation. For text generation, we studied the effects of the different sampling parameters. We found the best results in terms of text diversity and coherence through manual tuning with the following value combinations: number of beams: 1, beam size: 30, nucleus sampling value: 0.8, temperature: 2.0, and n-gram repetition limit: 3.

4.4 Measures

To evaluate the performance on the downstream task, we use accuracy. Training and inference time were also measured. In the *DP* setup, we measure privacy preservation with the privacy budget ϵ. In all models, the extent of unintended memorization of named entities found in the fine-tuning dataset is measured by counting their occurrences in generated samples and checking their k-eidetic value. A data point (or in our case an entity) is k-eidetic if it appears k times in the training corpus [8]. In particular this means that a 1-eidetic data point was observed by a model only exactly once.

5 RESULTS

Given the pipeline and experimental apparatus described in Sections 3 and 4, we evaluate the results of the classification performance and the extent of unintended memorization of each model. We compare our findings on both datasets.

5.1 Classification

Table 3 shows the singe-label text classification results for each setup. For both datasets, we observe a similar trend between the different fine-tuning setups: *Full* achieved the highest accuracy on the test set, *Partial* performed slightly worse, and the *DP* setup produced the worst results with a 15 percent point drop compared to the *Partial.* In general, the accuracy values are considerably higher on the Enron dataset. In the *DP* setup, the privacy budget ϵ is 9.79 for the Enron and $\epsilon = 7.38$ for Blog Authorship.

Table 3: Accuracy scores after fine-tuning

Fine-tuning Setup	Enron	Blog Authorship
Full	87.5%	51.6%
Partial	85.8%	49.6%
DP	69.8%	35.7%

Runtime Performance of Fine-tuning. We provide the total training times of the experiments in Table 4 executed on a single NVIDIA A100-SXM4-40GB card. *Full* fine-tuning is the slowest due to the fact that optimization happens for all layers. Comparing the two setups when only the last layers are optimized, *DP* fine-tuning takes notably longer time than *Partial*, but is still faster than the *Full* setup.

Table 4: Training runtime in minutes. Averaged over five runs.

Fine-tuning Setup	Enron	Blog Authorship
Full	11	330
Partial	6	186
DP	9	222

5.2 Named Entity Memorization

For the named entity memorization experiments, we also included a purely pre-trained BERT, i. e., without any fine-tuning, which we call the *Base* setup. Figure 3 shows our initial results on the *All* entity set. The highest extraction rate was 9.3% for the Enron and 6.2% for the Blog Authorship dataset. On the Enron dataset, the highest extraction rate was achieved on the *Base* setup. closely followed by the *Partial* setup. The difference between these two setups was 0.2% with the naive prompting and 0.6% for the informed prompting methods. On the Blog Authorship dataset, the *Full* setup produced the highest extraction rate, followed by the *Base* setup. Between these two setups, the naive prompting resulted in 0.5% and the informed prompting in a 2.4% difference. The *DP* setup produced the lowest extraction rates with 1.4% (naive prompting) and 1.1% (informed prompting) on the Enron and 0.1% in the Blog Authorship dataset (both, naive and informed prompting). Naive prompting consistently outperformed informed prompting in all fine-tuning setups on both datasets.

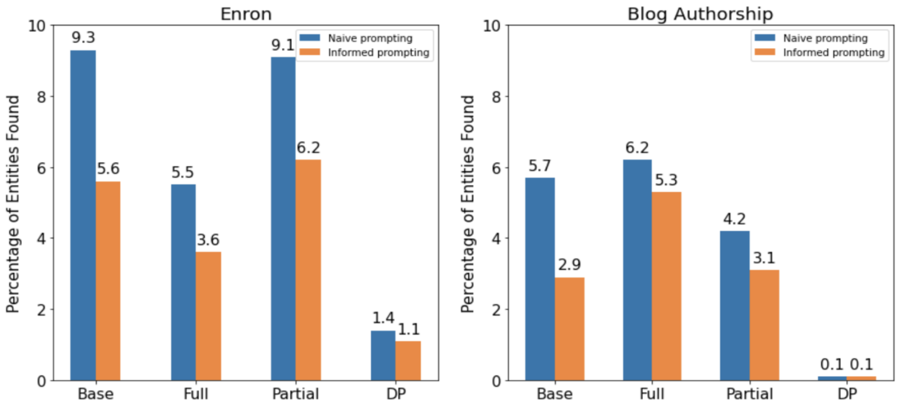

Figure 3: The percentages of all entities successfully extracted from the models, compared by prompting methods. In each column pair the left bar displays naive prompting and the right bar the informed prompting. The percentages are calculated based on the numbers shown in Table 2.

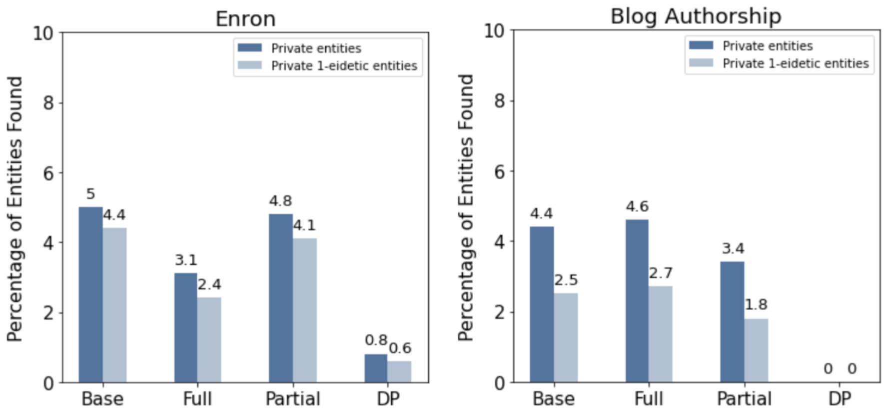

Figure 4: The percentages of private entities and private 1-eidetic entities successfully extracted from the models with the use of naive prompting. In each column pair, the left bar displays the results for the *Private* entity set and the right bar shows the results for the *Private* 1*-eidetic* entity set. The percentages are calculated based on the numbers shown in Table 2.

Table 5: Extraction ratio of entities from the *Private* set using naive prompting, grouped by entity types

Named Entity Type	Enron				Blog Authorship			
	Base	Full	Partial	DP	Base	Full	Partial	DP
PERSON	4.1%	2.3%	4.3%	0.8%	3.4%	4.1%	2.9%	*
ORG	3.8%	2.2%	3.3%	0.3%	4.5%	4.1%	3%	*
LOC	20.5%	15.4%	18.4%	5.1%	8.9%	8.6%	5.5%	*
GPE	28.1%	22.5%	28%	5%	11.5%	13.4%	9.9%	0.1%
FAC	1.7%	0.4%	0.8%	0.8%	2.7%	2.5%	1.6%	*
MONEY	1.5%	0.7%	1%	0.1%	1.2%	0.9%	0.7%	*
CARDINAL	4.8%	1.7%	3.9%	0.5%	4.4%	3.9%	2.7%	0.1%

* less than 0.1%

Figure 4 shows the comparison of the extraction rates between the private entities and the private 1-eidetic entities, using the naive prompting method. Compared to the results in Figure 3, the extraction rates are consistently lower across all setups. The difference between *Base* and *Partial* on Enron, and *Base* and *Full* on Blog Authorship once again is negligible. Overall the memorization rate of private 1-eidetic entities is lower than the memorization rate of all private entities. But the difference is less than 1 percent point on the Enron and less than 2 percent points on the Blog Authorship dataset.

To further investigate the extracted private entities, we also measured the extraction ratio of each entity type in Table 5. The Location and Geo-Political Event types produced the highest percentages, while the Facility and Money types had results less than 3% across all setups and datasets. The extraction ratios on Blog Authorship are consistently lower in every entity type compared to Enron. The only exception can be seen in the Facility type, where the Blog Authorship results were 1 to 2 percent points higher.

Additional plots of the results can be found in Appendix B.2. Examples of text generated by the three fine-tuning setups are provided in Appendix B.3.

6 DISCUSSION

6.1 Key Insights

Prompting Methods. Our experiments show that the naive prompting method produces better results in all setups. Although for informed prompting the seed sequences will be more similar to the text sequences found in the fine-tuning data, this informed prompting likewise limits the possibilities of producing diverse outputs. Following Carlini et al. [8], we conclude that using random prompts sampled from a huge corpus unrelated to the training data yields better extraction results. This shows that adversaries do not need to have prior knowledge about the training data of the attacked model, a simple black-box approach is sufficient.

Named Entity Memorization in BERT. We extracted private named entities from the fine-tuned models at surprisingly low rates. In no setup, we extracted more than 10% of the private entities. Interestingly, our results further show that using a pre-trained *Base* model that has not been fine-tuned on the training set containing those extracted entities produces similar extraction ratios. Our assumption is that the small percentage of private entities that have been successfully extracted from both the *Base* and *Full* or *Partial* models have low level of complexity in terms length and n-gram diversity. Therefore they are more likely to be randomly generated by combining common subword tokens.

In order to better understand the reasons behind this observation, we conducted a more detailed analysis of the extracted entities. As can be seen from Table 5, distinct entity types have different probabilities to be extracted. From the seven types, we argue Location and Geo-Political Event are the least unique in their nature, therefore it is not suprising that the highest extraction rates have been achieved on them. The lower values in the Money and Cardinal types reinforce the findings that the subword tokenization in BERT is a suboptimal method to encode numerical values [69]. Overall our findings suggest that BERT could be rather resistant to training data extraction attacks, unlike other large LMs, such as GPT-2 [8]. This is most likely due to its relatively smaller size as argued in [8], but it is also possible that auto-encoder transformers are generally less prone to these attacks compared to auto-regressive transformers as result of their different pre-training objectives.

Differentially Private Fine-tuning. In all the experiments that used Differentially Private fine-tuning, the extraction rates of named entities were reduced by a large extent. Our samples have shown that the text quality in the *DP* setups was very low, both text coherence and text diversity decreased dramatically. Even though the performance on the downstream task was also considerably lower, we argue this trade-off between performance and privacy is still promising for future developments. Considering that the focus of our study was not on achieving state-of-the art performance for singe-label text classification, we only used the Adam variant of the original DPSGD algorithm [1]. More recent DP algorithms retain a higher performance on many NLP tasks, while achieving the same or better privacy budgets [76]. One can expect that for tasks, where the ability of a model to generate text is irrelevant, the use of DP can be a viable solution to increase the privacy of the model.

6.2 Generalization

In our experiments, we intentionally used datasets of different characteristics. While the Enron dataset we used is small in size and is very cluttered due to its source (real world emails), the Blog Authorship Corpus is a per-default public web corpus that contains a large amount of samples covering a broad range of domains with a higher text quality. Although, we only used single-label text classification (in which BERT is generally considered as state-of-the-art [23]) as a downstream task for fine-tuning, results should be similar on different downstream tasks since the memorization takes place in the encoding layers, irrespectively of the task-specific final layers of a model. Finally our conclusion about the memorization capabilities of the BERT base model is in line with the training data extraction study done on Clinical BERT, in which the authors were unable to reliably extract patient names from a specific BERT variant pre-trained on clinical data [37].

6.3 Threats to Validity

We acknowledge that the experimental datasets are limited to English. Although named entities are often unique to their respective language, we have no reason to believe that generating named entities would be significantly easier in other languages. For languages that have larger character sets (e. g., Chinese) or use long compound words (e. g., German), the probability of unintended memorization may even be smaller. Regarding the efficiency of our extraction of named entities, the results can be influenced by both the named entity recognition system and our text generation method. Although we used the state-of-the-art named entity recognition model from spaCy, it is highly possible that some entities have been missed and some have been falsely identified. The missed entities are unlikely to influence the results since we still had a great amount of entities of differing k-eidetic values. Controlling for the falsely identified entities was a more difficult problem. Therefore, we decided to remove all entities with a character length of less than 4. While this does not solve the issue completely, we assume that the remaining false positives did not hinder the study of unintended memorization. Using a left-to-right sequential text generation method (based on the naive/informed prompting) might also bias our results, as BERT uses context from both directions to predict a token during pre-training. This, we argue has more impact on text coherence rather than the ability to trigger a diverse output containing named entities, which was of higher importance to our study.

6.4 Practical Impact and Future Work

Our study offers insights into the privacy risks involved in employing a pre-trained BERT on private fine-tuning data. Although our results do not rule out the possibility to extract personal information from a fine-tuned BERT base model using more advanced methods, our findings suggest that doing so is at least not trivial. As for future work it would be interesting to re-run the experiments on BERT large and other commonly used BERT variants [34, 41, 59]. Even though our experiments revealed surprisingly minimal privacy leakage in BERT models fine-tuned by standard methods, it would be interesting to analyze the effects of Differential Privacy in other language models. For example, it is known that GPT-2 is strongly prone to training data extraction attacks [8]. Another interesting area of future research would be to test the embedding layers of our BERT setups against membership inference attacks [44].

7 CONCLUSION

We performed an initial investigation into the capabilities of BERT to memorize named entities. We ran experiments, in which we tried to extract private named entities from fine-tuned BERT models using three different fine-tuning methods and two prompting strategies. Overall, we could only extract a low percentage of named entities from BERT, and the only pre-trained model generates the same amount entities as the fine-tuned models. Furthermore, we show that a fine-tuned BERT does not generate more named entities specific to the fine-tuning dataset than a BERT model that is pre-trained only. This does not rule out that PPI extraction from a fine-tuned BERT base model is impossible, but at least suggests that it is not trivial to emit personal or privacy sensitive named entities via training data extraction attacks. We also employed a Differential Private fine-tuning method, which showed to be a promising privacy preserving method against training data extraction attacks.

To facilitate further research, we make our experimental setups and baseline models available: https://github.com/drndr/bert_ent_attack

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–35.

[3] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323* (2019).

[4] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. 2017. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security* 13, 5 (2017), 1333–1345.

[5] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. [n.d.]. Language models are few-shot learners. *Advances in neural information processing systems* ([n. d.]).

[7] 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)*. 267–284.

[8] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

[9] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter* 19, 2 (2017), 25–35.

[10] Ali Davody, David Ifeoluwa Adelani, Thomas Kleinbauer, and Dietrich Klakow. 2020. Robust differentially private training of deep neural networks. *arXiv preprint arXiv:2006.10919* (2020).

[11] Emiliano De Cristofaro. 2020. An overview of privacy in machine learning. *arXiv preprint arXiv:2005.08679* (2020).

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[13] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305* (2020).

[14] 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* (2021).

[15] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 486–503.

[16] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407.

[17] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[18] Wikimedia Foundation. [n.d.]. *Wikimedia Downloads*. https://dumps.wikimedia.org

[19] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.

[20] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *23rd USENIX Security Symposium (USENIX Security 14)*. 17–32.

[21] Markus Freitag and Yaser Al-Onaizan. 2017. Beam Search Strategies for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*. 56–60.

[22] Lukas Galke and Ansgar Scherp. 2022. Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide MLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4038–4051.

[23] Lukas Galke and Ansgar Scherp. 2022. Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 4038–4051. https://doi.org/10.18653/v1/2022.acl-long.279

[24] Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), 721–741.

[25] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324* (2019).

[26] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*. PMLR, 201–210.

[27] Fadi Hassan, Josep Domingo-Ferrer, and Jordi Soria-Comas. 2018. Anonymization of unstructured data via named-entity recognition. In *International conference on modeling decisions for artificial intelligence*. Springer, 296–305.

[28] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* (2019).

[29] Matthew Honnibal, Ines Montani, Sofie Van Landeghe, and Adriane Boyd. 2022. spaCy: Industrial-strength Natural Language Processing in Python". https://zenodo.org/record/121230

[30] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[31] Ting Jiang, Deqing Wang, Leilei Sun, Zhongzhi Chen, Fuzhen Zhuang, and Qinghong Yang. 2022. Exploiting Global and Local Hierarchies for Hierarchical Text Classification. *CoRR* abs/2205.02613 (2022). https://doi.org/10.48550/arXiv.2205.02613 arXiv:2205.02613

[32] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[33] Bryan Klimt and Yiming Yang. 2004. Introducing the Enron Corpus. In *CEAS 2004 - First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA*. http://www.ceas.cc/papers-2004/168.pdf

[34] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[36] Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would Elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090* (2019).

[37] Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C Wallace. 2021. Does BERT Pretrained on Clinical Notes Reveal Sensitive Data? *arXiv preprint arXiv:2104.07762* (2021).

[38] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clement Delangue, Theo Matussiere, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, Francois Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 175–184.

[39] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–36.

[40] Kin Sum Liu, Bo Li, and Jie Gao. 2018. Generative model: Membership attack, generalization and diversity. *CoRR, abs/1805.09898* (2018).

[41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[42] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence.* 4513–4519.

[43] Zihan Liu, Genta Indra Winata, Andrea Madotto, and Pascale Fung. 2020. Exploring fine-tuning techniques for pre-trained cross-lingual models via continual learning. *arXiv preprint arXiv:2004.14218* (2020).

[44] Saeed Mahloujifar, Huseyin A Inan, Melissa Chase, Esha Ghosh, and Marcello Hasegawa. 2021. Membership Inference on Word Embedding and Beyond. *arXiv preprint arXiv:2106.11384* (2021).

[45] Huanru Henry Mao. 2020. A survey on self-supervised pre-training for sequential transfer learning in neural networks. *arXiv preprint arXiv:2007.00800* (2020).

[46] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629* (2016).

[47] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP).* IEEE, 691–706.

[48] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmaeilzadeh. 2020. Privacy in deep learning: A survey. *arXiv preprint arXiv:2004.12254* (2020).

[49] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.

[50] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, Yoav Goldberg and Stefan Riezler (Eds.). ACL, 280–290. https://doi.org/10.18653/v1/k16-1028

[51] Seong Joon Oh, Bernt Schiele, and Mario Fritz. 2019. Towards reverse-engineering black-box neural networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning.* Springer, 121–144.

[52] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755* (2016).

[53] Mathias PM Parisot, Balazs Pejo, and Dayana Spagnuelo. 2021. Property Inference Attacks on Convolutional Neural Networks: Influence and Implications of Target Model's Complexity. *arXiv preprint arXiv:2104.13061* (2021).

[54] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. OpenAI. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[55] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[56] Maria Rigaki and Sebastian Garcia. 2020. A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646* (2020).

[57] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how bert works. *Transactions of the Association for Computational Linguistics* 8 (2020), 842–866.

[58] Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. 2009. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *arXiv preprint arXiv:0911.5708* (2009).

[59] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[60] J Schler, M Koppel, S Argamon, and JW Pennebaker. 2006. Effects of Age and Gender on Blogging in Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs. In *Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs.*

[61] Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training NLP models: A concise overview. *arXiv preprint arXiv:2004.08900* (2020).

[62] 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).* IEEE, 3–18.

[63] Fabian Singhofer, Aygul Garifullina, Mathias Kern, and Ansgar Scherp. 2021. A novel approach on the joint de-identification of textual and relational data with a modified mondrian algorithm. In *DocEng '21: ACM Symposium on Document Engineering 2021, Limerick, Ireland, August 24-27, 2021*, Patrick Healy, Mihai Bilauca, and Alexandra Bonnici (Eds.). ACM, 14:1–14:10. https://doi.org/10.1145/3469096.3469871

[64] Weiqi Sun, Haidar Khan, Nicolas Guenon des Mesnards, Melanie Rubino, and Konstantine Arkoudas. 2022. Unfreeze with Care: Space-Efficient Fine-Tuning of Semantic Parsing Models. In *Proceedings of the ACM Web Conference 2022.* 999–1007.

[65] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.

[66] Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Françoise Beaufays. 2020. Understanding unintended memorization in federated learning. *arXiv preprint arXiv:2006.07490* (2020).

[67] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *25th USENIX security symposium (USENIX Security 16).* 601–618.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[69] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* 5307–5315.

[70] Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a markov random field language model. *arXiv preprint arXiv:1902.04094* (2019).

[71] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy (SP).* IEEE, 36–52.

[72] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.

[73] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations.* 38–45.

[74] Yang Xu, Tinghuai Ma, Meili Tang, and Wei Tian. 2014. A survey of privacy preserving data publishing using generalization and suppression. *Applied Mathematics & Information Sciences* 8, 3 (2014), 1103.

[75] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. 2021. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298* (2021).

[76] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2021. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500* (2021).

[77] Santiago Zanella-Beguelin, Lukas Wutschitz, Shruti Tople, Victor Ruhle, Andrew Paverd, Olga Ohrimenko, Boris Kopf, and Marc Brockschmidt. 2020. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security.* 363–375.

[78] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337* (2022).

[79] Jiajun Zhang, Chengqing Zong, et al. 2015. Deep Neural Networks in Machine Translation: An Overview. *IEEE Intell. Syst.* 30, 5 (2015), 16–25.

[80] Tianwei Zhang, Zecheng He, and Ruby B Lee. 2018. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860* (2018).

[81] Jingwen Zhao, Yunfang Chen, and Wei Zhang. 2019. Differential privacy preservation in deep learning: Challenges, opportunities and solutions. *IEEE Access* 7 (2019), 48901–48911.

[82] Tianqing Zhu, Dayong Ye, Wei Wang, Wanlei Zhou, and Philip Yu. 2020. More than privacy: Applying differential privacy in key areas of artificial intelligence. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[83] Yukun Zhu, Ryan Kiros, Rich 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 *The IEEE International Conference on Computer Vision (ICCV).*

Supplementary Materials

A EXTENDED DATASET PREPROCESSING

The following sections contain additional details about the preprocessing. In particular, we give further information about the preparation of the Enron dataset, and provide an overview of the label distributions.

A.1 Preprocessing the Enron dataset

As mentioned in Section 4.1, the Enron dataset does not come with categorical labels. The raw dataset contains the text of the email messages, alongside additional headers with meta-information. One of these headers called "X-folder" contains the location of the email in folder structure of each user. We used the last folder name in the location paths to create labels. While these folders are unique to each user, some general overlap between the naming conventions and folder contents made it possible to select seven folders as labels that can be used for single-label text classification. These labels are: "logistics", "personal", "management", "deal discrepancies", "resumes", "online trading", and "corporate".

While preprocessing the text of the emails, we found that the message bodies contained a great amount of message chains, where some meta-information from the email headers (such as lists of email addresses, network information, message ID) were also present between the texts. This turned out to have a negative effect on output of our models when generating text. Namely, the generated samples lost all coherence and contained a lot of random concatenations of subword tokens. Therefore, we decided to discard all reply and forward chains from the emails by removing the parts following phrases that indicate a replied or forwarded message. This additional preprocessing, alongside the removal of HTML links, substantially improved the diversity and quality of the generated text samples.

A.2 Label Distributions

Figures 5 and 6 show the label distribution of both datasets. In the Enron dataset, the most frequent label is "personal" with 2,062 occurrences, while on the Blog Authorship dataset the label "Student" has the highest count with 153,903. During the train-test split, the distribution of the dataset was retained in both splits.

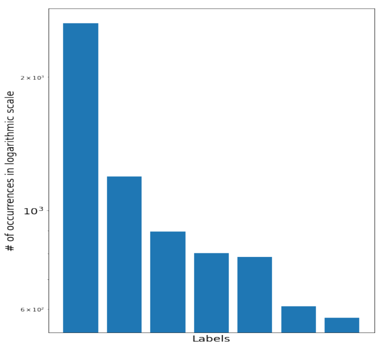

Figure 5: Label distribution of the Enron dataset

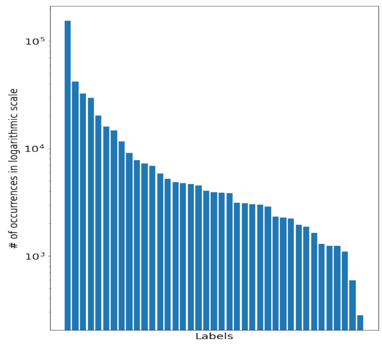

Figure 6: Label distribution of the Blogauthorship dataset

B EXTENDED EXPERIMENT RESULTS

This section contains the extended experimental results. We provide an extended review of the NER results with detailed description of the entity types, a discussion about selecting specific entity types for our study, and the additional named entity extraction results not present in the main paper.

B.1 Named Entity Recognition

As mentioned in Section 4.2.3, we used the spaCy library [29] for the process of NER. The named entity recognizer of spaCy distinguishes between the following 18 entity types:

- **PERSON** - people names, including fictional
- **NORP** - nationalities or religious or political groups
- **FAC** - facilities - building, airports, bridges, etc.
- **ORG** - organizations, companies, agencies, institutions
- **GPE** - geopolitical entities - countries, cities, states
- **LOC** - non-GPE locations
- **PRODUCT** - objects, vehicles, foods
- **EVENT** - named hurricanes, wars, sport, events etc
- **WORK_OF_ART** - titles of books, songs, etc
- **LAW** - named documents made into laws
- **LANGUAGE** - any named language
- **DATE** - absolute or relative dates, periods
- **TIME** - times smaller than a day
- **PERCENT** - percentages
- **MONEY** - monetary values, including unit
- **QUANTITY** - Measurements, as of weight or distance
- **ORDINAL** - "first", "second", etc
- **CARDINAL** - numerical values not covered by other types

The initial results of the NER can be seen in Table 6. Compared to the values in Table 2, these counts also include the repeated occurrences of a named entity.

In studying the extent of named entity memorization, we focused on entity types that have a higher probability to contain personal or privacy sensitive information. *PERSON* can include first and last names, which together can be considered as personal information. *ORG*, *GPE*, *LOC* and *FAC* include information that can be pieced together to identify a likely data subject, therefore they also fit into the personal information category. We included *Money*, as in some industries (i. e., banking) records of specific amounts can be regarded as privacy sensitive information. Finally, we also included *CARDINAL* as this type can refer to card numbers, phone numbers, and different ID numbers, which can be both personal and privacy sensitive.

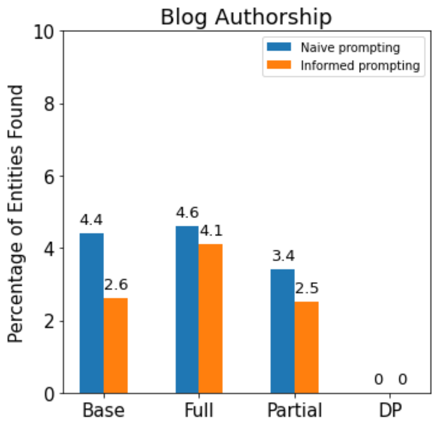

Figure 8: The percentages of private entities extracted from the models, compared by prompting method.In each column pair the left bar displays naive prompting and the right bar the informed prompting. The percentages are calculated based on the numbers shown in Table 2.

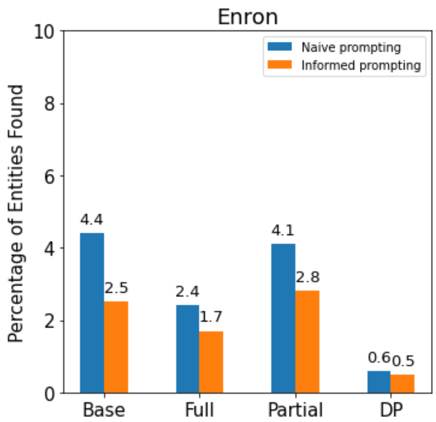

Figure 9: The percentages of private 1-eidetic entities extracted from the models, compared by prompting method.In each column pair the left bar displays naive prompting and the right bar the informed prompting. The percentages are calculated based on the numbers shown in Table 2.

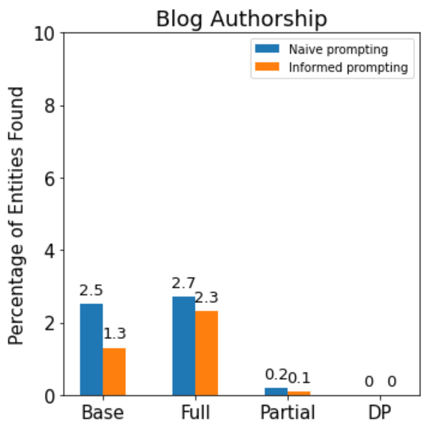

Figure 10: The percentages of private entities extracted from the models, compared by prompting method. In each column pair the left bar displays naive prompting and the right bar the informed prompting. The percentages are calculated based on the numbers shown in Table 2.

Table 6: Per type number of named entities in the datasets

Named Entity Type	Enron	Blog Authorship
PERSON	33,993	767,144
NORP	1,373	132,080
FAC	674	24,405
ORG	30,365	521,159
GPE	8,771	298,607
LOC	827	37,286
PRODUCT	992	27,432
EVENT	333	14,131
WORK_OF_ART	1,354	43,102
LAW	151	5,798
LANGUAGE	69	11,161
DATE	17,319	626,969
TIME	7,137	211,993
PERCENT	579	17,332
MONEY	2,952	39,768
QUANTITY	655	21,512
ORDINAL	1,210	104,559
CARDINAL	17,725	444,764

B.2 Named Entity Memorization

This section includes the results of the experimental setups not shown in the main text. The percentage values are based on the number shown in Table 5.

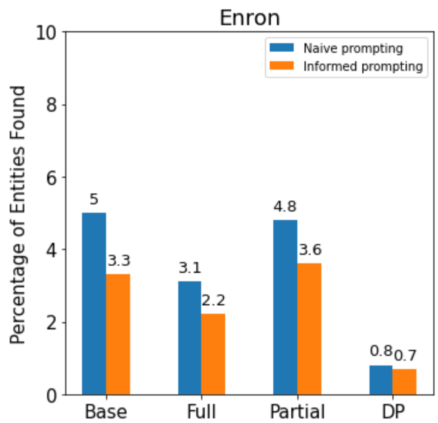

Figure 7: The percentages of private entities extracted from the models, compared by prompting method. In each column pair the left bar displays naive prompting and the right bar the informed prompting. The percentages are calculated based on the numbers shown in Table 2.

B.3 Text Generation

This section includes text samples generated from our fine-tuned models. Tables 7, 8, and 9 show randomly selected sequences of generated text under the *Full*, *Partial*, and *Differentially Private* fine-tuning setting for each of the two datasets and prompting strategies.

Table 7: Randomly selected sequences of generated text from the *Full* setup

Generated Sample	Dataset	Prompt method
Organizations from among 150 organizations and national educational leaders Global Quality Internet Service Vision Vision Watch Associates Association National / CT Science Vision Health network	Enron	naive
In, Justice & Justice : International / Research Papers Issues to 2017. : International Organizations Research in Business Fellow and Masiah Al Hehar and Professor and Chair, University and Karachi Emeritus, from Durham	Enron	informed
Third to the Joint Chiefs, between 2015 - 2020 side directors permanent appointment playing two leaders joint in training urban core areas in economic and medical development organisations in electoral the 2020 European Parliament	Blog Authorship	naive
Sheriff Avery and Judge said everyone and his girls were safe and they all was riot! Only Judge White and the churchmaster had not heard tomorrow, and things was right Maydal Sio and all his lawyers were punishment	Blog Authorship	informed

Table 8: Randomly selected sequences of generated text from the *Partial* setup

Generated Sample	Dataset	Prompt method
Annex 1 at Anne. M. 1, featuring new entrances and entrance towards the entrances from East Village and West Village Parkway (connecting to and from both tunnels into and from West Village) also includes the entire structure encompassing the new tunnels on all four floors including the main floors plus all 4 rooms	Enron	naive
Sir Christopher and to Commander Fox, presented a statement that they owed to Saunders for saving Christopher before sailing for Britain. Saunders blamed Christopher ; they were wrong, obviously not, and clearly angered they were, but Adams insisted Christopher instead was " damaged, was very " sick and " emotionally "	Enron	informed
Those last day. Those those strange voices - Those eerie voices - That haunted voices - that haunting voice, That haunting ghost - that echoing in echoes of the haunting? Why silence? Why? - The ghost? Which is it? who is ghost And silent voice?	Blog Authorship	naive
Paul. Paul leaves away before that scene breaks as Helen laughs bitterly with Julia and Liz. Paul hates love and they hate him because they love him so bad so love, her toils against Paul and hasces with Paul. Paul makes love Simon	Blog Authorship	informed

Table 9: Randomly selected sequences of generated text from the *Differentially Private* setup

Generated Sample	Dataset	Prompt method
exampleenerMFuzkt few1 thateg5 talkden landkin peopledan mentionilialiltendan1tagB landseB ofsemadlan mouth talkrodrodelinetieei mouth mouthmadpenukensorlattensorelepedlen start1ndimilebuuzseuzalaeleicidanici mouthisipen	Enron	naive
mouthsixhilueriknt onhillattlattedelenacherwearelinestlubchorileelinsionenerpenc ionpenkineonelinvilsettsablepencardtinxinpensablelansorN-mileelinbenkinkinndiglielinkinxinpenpenelinchinelipeth"	Enron	informed
,. " in him not guitar around them no brother was time and no The up lap up wasly guitar no againul un from without lap aside " sex of time as " The if them a no? " aside all around aside around un.. down t O un from guitar aside without without O O him aside " No A " in time brother guitarist unul The The in„ him was out with.	Blog Authorship	naive
lap in No the The while roundul guitar no was the guitarcase without was aside the " without A without t guitars " round while, o team as like draw O O as done small o up not was and and out out away in in. the aside without guitarist., until t the team away. at down no not up down " if t up the guitar as of the. in away, un away	Blog Authorship	informed