Preserving Privacy in Large Language Models: A Survey on Current Threats and Solutions

Anonymous authors
Paper under double-blind review

Abstract

Large Language Models (LLMs) represent a significant advancement in artificial intelligence, finding applications across various domains. However, their reliance on massive internet-sourced datasets for training brings notable privacy issues, which are exacerbated in critical domains (e.g., healthcare). Moreover, certain application-specific scenarios may require fine-tuning these models on private data. This survey critically examines the privacy threats associated with LLMs, emphasizing the potential for these models to memorize and inadvertently reveal sensitive information. We explore current threats by reviewing *privacy attacks* on LLMs and propose *comprehensive solutions* for integrating privacy mechanisms throughout the entire learning pipeline. These solutions range from anonymizing training datasets to implementing differential privacy during training or inference and machine unlearning after training. Our comprehensive review of existing literature highlights ongoing challenges, available tools, and future directions for preserving privacy in LLMs. This work aims to guide the development of more secure and trustworthy AI systems by providing a thorough understanding of privacy preservation methods and their effectiveness in mitigating risks.

1 Introduction

Large Language Models (LLMs) represent a significant advancement in artificial intelligence (AI), demonstrating remarkable capabilities in understanding and generating human-like text, and fueling innovation across various domains (Brown et al., 2020; Touvron et al., 2023; Zhao et al., 2023). However, their reliance on massive internet-sourced datasets for training raises significant privacy concerns considering that these models are prone to memorize segments of their training data (Feldman, 2020; Feldman & Zhang, 2020). This is further exacerbated if we consider certain application-specific scenarios (e.g, healthcare) that may requires the fine-tuning of publicly available models on user private data (Tramèr et al., 2022). Since language models are specifically trained to generate text, this poses a fundamental threat considering that they might inadvertently disclose memorized private data at any moment (Carlini et al., 2019; 2021).

Privacy is a fundamental human right that encompasses the protection of personal information from unauthorized access and misuse (United Nations, 1948; Nissenbaum, 2010). Preserving privacy in the context language language models thus means implementing measures that ensure the protection of personal and sensitive information from being exposed or misused, following the permissions granted by users to use those data, whether implicit or explicit (Brown et al., 2022a). Preserving privacy is not only crucial to preserve users and public trust in AI technologies, but also a legal requirement in many countries such as the EU with the General Data Protection Regulation (GDPR), the US with the Privacy Act, and China with the Personal Information Protection Law and Data Security Law (Weidinger et al., 2021; Solaiman et al., 2019; Veale et al., 2018). These regulations may include additional guarantees, such as the "right to be forgotten", which allows individuals to withdraw their consent and request the deletion of their personal data, posing an additional challenge for LLMs as erasing a data point from a trained model might require retraining the model from scratch or discarding the entire model (Zhang et al., 2024b).

Understanding the distinction between privacy and copyright is crucial in this context. Privacy focuses on the protection and control of personal information, ensuring that individuals' sensitive data remains confidential

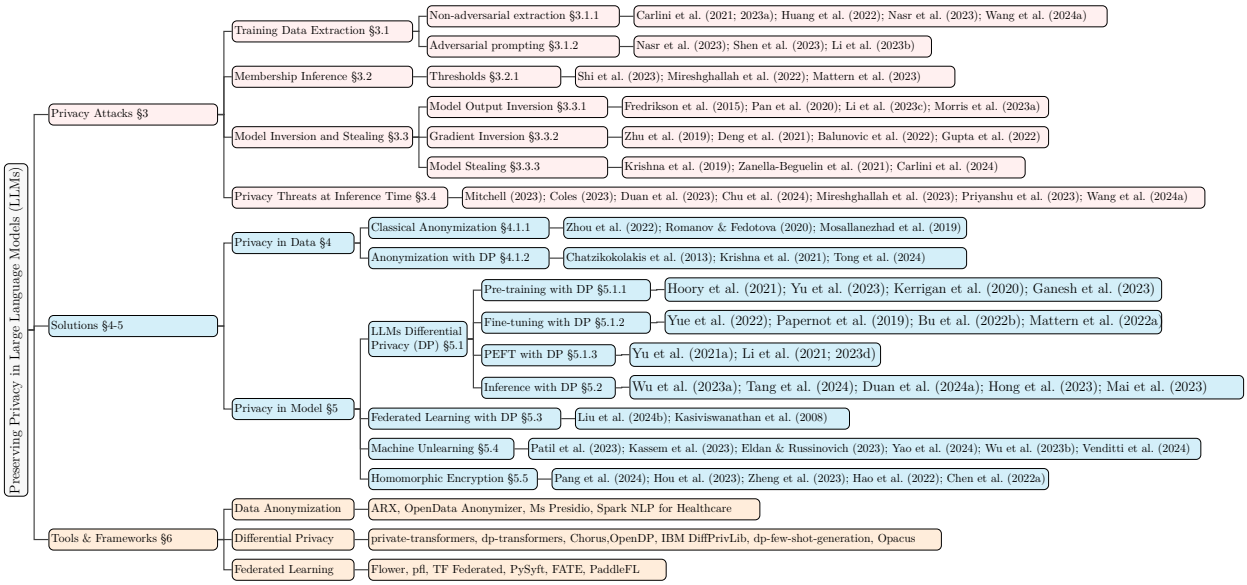

Figure 1: **Taxonomy of Preserving Privacy in Large Language Models.** Red indicates various attack techniques, and blue represents current possible solutions to preserve privacy by acting on the training data or model. Finally, in orange we highlight the currently available tools to preserve privacy.

and is not shared or exploited without their consent. Copyright, on the other hand, pertains to the legal rights of creators over their original works, such as text, music, or art, and governs how these works can be used, distributed, and reproduced by others (Torremans, 2007; Karamolegkou et al., 2023). While privacy is about protecting personal data from unauthorized access, copyright deals with protecting intellectual property and ensuring that right holders receive recognition and compensation for their work. Our focus in this study is on addressing privacy issues related to the potential exposure of personally identifiable information (PII) in LLMs, rather than exploring copyright infringement or intellectual property rights.

This survey aims to provide an holistic view on the problem of preserving privacy in LLMs by combining both the current *privacy threats* coupled with the current *available solutions* and tools to implement them. We believe this approach, which presents both aspects in tandem, to be an essential resource for researchers and practitioners seeking a comprehensive understanding of this field. Specifically this survey targets: (i) readers interested in understanding the current privacy risks associated with LLMs (§3); (ii) readers seeking to learn about the available solutions to defend against privacy attacks (§4 - 5); and (iii) readers looking for tools to make their LLMs resistant to privacy attacks (§6).

PII

Personal Identifiable Information (PII) refers to any data that can be used to identify a specific individual. This may include direct information such as a person's name, address, email address, social security number, phone number, genome, fingerprints, and face, as well as any other data that, when combined, could lead to the identification of an individual (National Institute of Standards and Technology, 2017; General Data Protection Regulation, 2016).

The survey is organized as follows and divided into two main categories: *privacy attacks* (§3) and *solutions* (§4 - 5). The taxonomy in Figure 1 provides a structured view of our survey on preserving privacy in LLMs. For readers unfamiliar with the terminology, we provide a preliminaries section (§2) that explains the basic concepts related to language models, differential privacy, and federated learning.

Privacy attacks discuss how its possible to extract training data from a model and encompass training data extraction (§3.1), membership inference (§3.2), and model inversion and stealing (§3.3). Training data extraction involves techniques such as non-adversarial extraction (§3.1.1) and adversarial prompting

(§3.1.2), where attackers can potentially extract sensitive information from the training data. Membership inference (§3.2) utilizes shadow models or thresholds (§3.2.1) to infer whether specific data points were part of the training set. Model inversion includes model output inversion (§3.3.1) and gradient inversion (§3.3.2) methods that can reconstruct the input data from model outputs while Model stealing is proposed to reconstruct a model's parameters from its output (§3.3.3). However, as prompting LLMs with private input – such as in-context demonstrations – is an increasingly popular approach, threats at inference time are also discussed 3.4.

Solutions to these privacy challenges are categorized into privacy in data (§4) and privacy in model (§5). Privacy in data includes classical anonymization techniques to anonymize data before it is used in training (§4.1.1) and anonymization with differential privacy (DP) (§4.1.2), which ensures that individual data points cannot be distinguished from the dataset. Privacy in model encompasses the application of DP during different phases such as pre-training (§5.1), fine-tuning (§5.1.2), Parameter-Efficient Fine-Tuning (PEFT) (§5.1.3), and inference (§5.2). Federated learning with DP (§5.3) is implemented to maintain privacy across distributed datasets, machine unlearning (§5.4) involves techniques to effectively remove specific data from trained models, and Homomorphic Encryption (§5.5) involves cryptographic methods that allow computation on encrypted data without decrypting it. Finally, we present available tools and frameworks to implement privacy in language models (§6).

In summary, this survey serves as a critical resource for researchers, and practitioners providing insights into the current privacy threats in LLMs and the currently available solutions to address these challenges. By systematically categorizing privacy issues and their corresponding mitigation strategies, this paper not only highlights the importance of preserving privacy in LLMs but also paves the way for future advancements in the field. All the papers in this survey are also available at the following GitHub repository[1].

2 Preliminaries

In this section, we present preliminary concepts essential for comprehending the subsequent content of the document. Readers already acquainted with these concepts may proceed directly to the next section. For a comprehensive introduction to the concepts of Machine Learning and Deep Learning we direct the reader to Bishop (2006) and Goodfellow et al. (2016).

In this survey, we use the term large language model (LLM) to indicate any model trained on broad textual data that can be adapted (e.g., via fine-tuning or in-context learning) to a wide range of downstream tasks (Bommasani et al., 2021). While the focus of the survey is mostly on generative LLMs (Zhao et al., 2023)—models that are capable of solving a broad set of tasks by generating text, such as GPT-3 and GPT-4—it is not uncommon to encounter smaller models like BERT in our discussion. We have chosen to include some of these models when we believe that their associated findings, both in terms of potential threats and proposed solutions, remain still relevant and generalizable to larger models nowadays.

2.1 Large Language Models

Large language models are deep learning systems trained on vast amounts of not-annotated text data to predict the likelihood of word sequences. Currently, most of the generative language models are trained in an autoregressive fashion, meaning they generate text by predicting each word sequentially based on the previous words in the sequence - usually from left to right. A model trained in this way is capable of generating and manipulating language effectively, which has been shown useful across a broad set of tasks and scenarios (Zhao et al., 2023). However, this was not always the case. Other popular techniques train LMs by masking certain words in a sentence and then predicting those masked words based on the surrounding context, like in the case of BERT (Devlin et al., 2019). These models are typically much smaller and less capable than modern LLMs (for instance, BERT-large has 340 million parameters). Language modeling is not a new task: the groundwork for it was laid in 1948 by Claude Shannon (Shannon, 1948), and the task was extensively investigated for decades, but the recent advances with the introduction of the Transformer architecture (Vaswani et al., 2023) and the rollout of GPT-3 in 2022 (Brown et al., 2020) shed new light

[1] anonymized

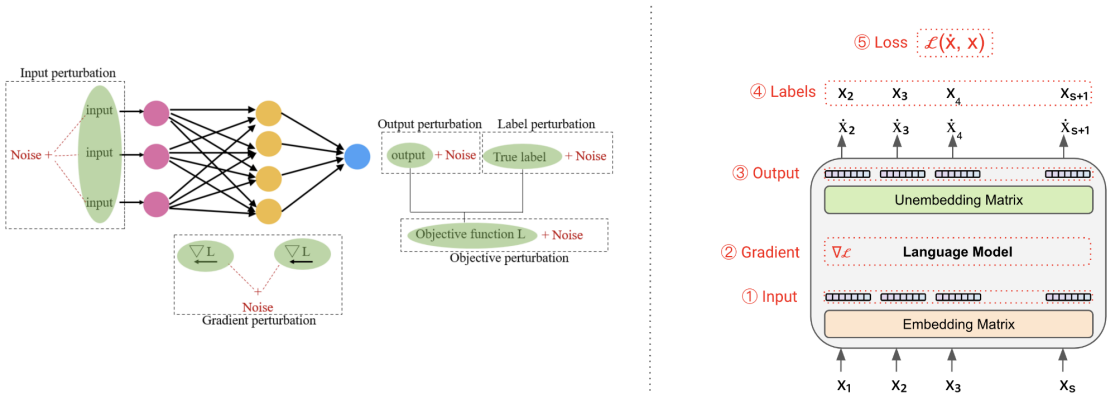

Figure 2: **Differential Privacy.** **One the left:** classical DP applied to a neural network, picture from Baraheem & Yao (2022). **On the right:** DP applied on a Language Model (analyzed in detail in Section 5.1). Differential privacy can be applied by adding a perturbation (noise) on different positions of its architecture: 1) **Input** noise perturbation is added to the input; 2) **Gradient** noise perturbation is added during training to the gradient update step; 3) **Output** noise perturbation is added to the output; 4) **Labels** noise perturbation is added to the labels used during training; 5) **Loss** noise perturbation is added to the loss function during training.

on the task and this new class of very big language models, namely large language models. Training a Large Language Model from scratch, sometimes referred to as pre-training, is usually very computationally intensive given the scale of these models and requires several resources generally available only to a few organizations in the world (Sharir et al., 2020; Hadi et al., 2023). For example, to give a sense of this scale, GPT-4 is rumored to have around 1.8 trillion parameters with an estimated training cost of around $63 million at the time of its release.

With these models, a new training paradigm arose: the pre-training/fine-tuning paradigm. Pre-training is usually on the next token prediction task, while fine-tuning is on the downstream task (which can still be the next token prediction, but it can also be something else like a simple classification task). Organizations building these large language models typically pre-train them on extensive text corpora and then fine-tune them for specific tasks before making them accessible to the public. White-box access allows users to download the model's weights and see or modify the model's internal parameters and architecture, enabling detailed inspection and customization. In contrast, black-box access is usually offered through APIs or web interfaces, where users can input data and receive outputs without any visibility into the model's internal workings or the ability to modify its parameters. There's also a middle ground called grey-box access, which allows users a partial view and limited interaction with the model's internal workings. Users can gain some insights into specific aspects of the model, such as the architecture or certain parameters, but do not have full access to download or modify the model's weights and internal components. This type of access might enable customization and tuning through pre-defined interfaces or tools, offering more flexibility and control than black-box access but not as much transparency or openness as white-box access. Researchers and scientists can efficiently adapt the models to their specific tasks and needs through fine-tuning, which uses a relatively small fraction of the original pre-training data and iterations. Since these are language models, it is possible to interact with them in natural language, by feeding them sentences in input, which in this context are called prompts. This means that black box models can still be adapted by conditioning them with specific prompts, either in a zero-shot or few-shot fashion (Brown et al., 2020). This process is known as in-context learning: in a zero-shot setting, the LLM takes only an instruction as input, while in a few-shot setting, it also takes a few examples, significantly enhancing its conditioning.

Here, we offer the reader a comprehensive explanation of several terminologies frequently used in the context of language models throughout the paper. These explanations aim to clarify key concepts and enhance understanding of the material presented:

- Sampling: LLMs are trained to predict the next token (word or punctuation) when given a text (prompt). To do so, LLMs learn a probability distribution over every possible token. Sampling consists of selecting the next word from this distribution, given the current context. Various methods, such as greedy sampling, random sampling, top-k sampling, and top-p sampling, allow for different balances between coherence and creativity in the output. These techniques are essential for tailoring the model's responses, impacting the diversity and appropriateness of text generation across applications like chatbots and content creation.

- Instruction tuning: Instruction tuning is a technique used to enhance the performance of large language models (LLMs) by training them to follow specific instructions. It involves fine-tuning a pre-trained model on a dataset that includes tasks described in natural language, along with the desired outputs. This technique allows the model to shift from general next-token prediction to more specialized tasks (e.g., text summarization). By using instruction tuning, the model learns to interpret and execute tasks based on user-given instructions, thereby improving its ability to generate relevant and accurate responses given a user prompt.

- PEFT (Parameter Efficient Fine-Tuning): Parameter Efficient Fine-Tuning is a class of methods to adapt pre-trained language models by altering only a small fraction of the model's parameters. This approach maintains the general capabilities of the model while enabling customization to new requirements with minimal computational cost and memory overhead. It's particularly useful for deploying models in resource-constrained environments or for applications requiring rapid adaptation to new data.

- RLHF (Reinforcement Learning from Human Feedback): Reinforcement Learning from Human Feedback is a technique used to train large language models (LLMs) by incorporating human preferences into the learning process. After a model has been instruction-tuned, its outputs are evaluated by humans, who provide feedback on the quality and relevance of the responses. This feedback is used to fine-tune a reward model that assigns scores to the outputs based on human preferences. The main model is then fine-tuned using reinforcement learning, guided by the reward model, to better align with human values and enhance its performance on tasks requiring nuanced understanding and judgment.

2.2 Differential Privacy

According to Dwork & Roth (2014): *"**Differential privacy** describes a promise, made by a data holder, or curator, to a data subject: "you will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available."* The fundamental goal of Differential Privacy is to enable the extraction of useful insights from data while minimizing the potential impact on any individual's privacy within that dataset. The core idea is to introduce controlled noise or randomness into the analysis or results in a way that prevents the identification of specific individuals. In simpler terms, when a system or algorithm is differentially private, the inclusion or exclusion of any individual's data should not significantly affect the overall outcome or conclusions of the analysis. There are several kinds of Differential Privacy definitions, but the original one is (ε, δ)-Differential Privacy.

(ε, δ)-Differential Privacy

A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

The formal guarantee provided by (ε, δ)-Differential Privacy is that for any two datasets x and x' that differ by only one element (these are often referred to as "adjacent datasets"), and for any subset of outputs S of the algorithm's range Y, the probability that the algorithm outputs something in S given x is not substantially higher than given x'.

ϵ is the parameter that measures the privacy loss associated with a query in the system. A smaller ϵ provides stronger privacy guarantees. The use of ϵ in the inequality indicates that the probability of a certain output does not increase by more than a multiplicative factor of e^{ϵ} when any single individual's data is added or removed.

The privacy budget is the total loss of privacy sustained by the system during the execution, and it can be computed by compounding the privacy loss indicated by ϵ over the total number of executions/iterations. This is done with the composition theorems, which are explained in more detail later in this section.

δ is the parameter that allows for a small probability of failure of the privacy guarantee. Ideally, δ should be very small, close to 0, which indicates a lower chance of the algorithm failing to maintain privacy. When δ is 0, it is defined ε-differential privacy, which is the strongest kind of differential privacy.

This means that for all subsets S of the output range Y, the probability that the algorithm on input x gives an output in S is at most e raised to the power ϵ times the probability that the algorithm on input x' gives an output in S, plus the small probability δ.

The simplest method to achieve differential privacy in a query is by adding random noise to its answer. The challenge lies in balancing the noise added to meet differential privacy standards without rendering the answer too noisy to be useful. To assist with this, foundational mechanisms have been developed in the field, detailing the type and amount of noise required. One such mechanism is the Laplace mechanism.

Laplace Mechanism

According to the Laplace mechanism, for a function $f(x)$ which returns a real number, the following definition of $F(x)$ satisfies ϵ-differential privacy:

$$F(x) = f(x) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right)$$

where Δf is the sensitivity of f, and $\text{Lap}(b)$ denotes sampling from the Laplace distribution with center 0 and scale b.

Sensitivity Δf refers to the maximum change in the output of a function or query when a single individual's data in the input dataset is added or removed. It quantifies how much a query's result can vary with the smallest possible change in the dataset, which is crucial in determining the amount of noise that needs to be added to ensure differential privacy. A function with high sensitivity requires more noise to protect individual privacy, while a function with low sensitivity requires less. Sensitivity is a key factor in calibrating the noise for mechanisms like the Laplace mechanism to ensure that individual data points do not significantly affect the output, thereby maintaining privacy.

Global Sensitivity

For a function $f : \mathcal{D} \to \mathbb{R}$ mapping datasets ($\mathcal{D}$) to real numbers, the global sensitivity of f is defined as follows:

$$GS(f) = \max_{x,x':d(x,x')<=1} |f(x) - f(x')|$$

Here, $d(x, x')$ denotes the distance between two datasets x and x', with the notion that two datasets are neighbors if their distance is 1 or less. The way this distance is defined significantly impacts the privacy definition we achieve, and we'll explore the dataset distance metric in more detail later.

Global sensitivity is defined such that for any two neighboring datasets x and x', the difference between $f(x)$ and $f(x')$ does not exceed $GS(f)$. This is referred to as "global" sensitivity because it is independent of the specific dataset being queried, applying to any pair of neighboring x and x'.

Based on the noise added, there are several mechanisms in the literature. The Gaussian mechanism is an alternative to the Laplace mechanism, which adds Gaussian noise instead of Laplacian noise.

Gaussian Mechanism

The Gaussian mechanism does not satisfy pure ϵ-differential privacy, but does satisfy (ϵ, δ)-differential privacy. According to the Gaussian mechanism, for a function $f(x)$ which returns a number, the following definition of $F(x)$ satisfies (ϵ, δ)-differential privacy:

$$F(x) = f(x) + \mathcal{N}(0, \sigma^2)$$
$$\text{where } \sigma = \frac{\Delta f \cdot \sqrt{2\log(1.25/\delta)}}{\epsilon}$$

where Δf is the sensitivity of f, and $\mathcal{N}(0, \sigma^2)$ denotes sampling from the Gaussian (normal) distribution with mean 0 and variance σ^2. Note that here (and elsewhere in these notes), log denotes the natural logarithm.

The primary mechanisms we've discussed (Laplace and Gaussian) are designed for numerical outputs, adding noise directly to the results. But what if we need to provide an exact answer without injecting noise yet still maintain differential privacy? The exponential mechanism addresses this by enabling the selection of the "best" element from a set while ensuring differential privacy. The "best" element is identified by using a scoring function that assigns scores to each item within a given set of options. The exponential mechanism ensures differential privacy by approximately maximizing the score of the chosen element. To preserve privacy, it might sometimes select an element that doesn't have the highest score.

Exponential Mechanism

The exponential mechanism satisfies ϵ-differential privacy:
1. A set $\mathcal{R}$ of possible outputs is selected
2. A scoring function is specified $u : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$ with global sensitivity Δu
3. The exponential mechanism outputs $r \in \mathcal{R}$ with probability proportional to:

$$\exp\left(\frac{\epsilon u(x, r)}{2\Delta u}\right)$$

The main practical distinction between the exponential mechanism and previous mechanisms like the Laplace mechanism is that the exponential mechanism's output is always an element from the specified set. This is particularly advantageous when choosing an item from a finite set, where a noisy result would be inappropriate.

Another prominent definition of Differential Privacy, which is actually a relaxation of the original definition, is Rényi Differential Privacy (Mironov, 2017).

Rényi Differential Privacy

A randomized algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is said to have ϵ-Rényi differential privacy of order α, or (α, ϵ)-RDP for short, if for any adjacent $D, D' \in \mathcal{D}$ it holds that:

$$D_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')) \leq \epsilon$$

Here, $D_\alpha(P\|Q)$ represents the Rényi divergence of order α between two probability distributions P and Q, and is defined as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log\left(\sum_{x \in \mathcal{X}} P(x)^\alpha Q(x)^{1-\alpha}\right)$$

for all $\alpha > 1$, where $\mathcal{X}$ is the support of the distributions P and Q. The parameter α here controls the sensitivity of the divergence measure to differences between P and Q, with higher values of α focusing on

events that are more probable under P than under Q. The parameter ϵ still quantifies the privacy guarantee, with smaller values indicating stronger privacy.

By providing a mathematically rigorous definition of privacy guarantees, Differential Privacy has become a key principle for designing privacy-preserving algorithms, especially in scenarios where personal data is involved. From this mathematical definition, some interesting properties arise. Among these, there are composition theorems that allow the computation of the exact privacy loss through many applications of the DP algorithm, which corresponds to the privacy budget. Sequential composition theorem (Kairouz et al., 2015) applies when multiple differentially private mechanisms are applied to the same dataset in sequence. The theorem quantifies how the privacy guarantees accumulate over the sequence of operations. Parallel composition theorem (McSherry, 2009) applies when multiple differentially private mechanisms are applied to disjoint subsets of the dataset. This theorem shows that the privacy guarantee of the overall process is governed by the worst-case privacy guarantee of any single mechanism. Differential Privacy also exhibits a Post-Processing property that states that once data has been processed through a DP mechanism, the output retains its privacy guarantees regardless of any further processing.

Differential privacy was born as a statistic analysis technique for databases. Initially, it was used to query databases without compromising the privacy of anybody enrolled, and that is why we discussed about queries and compounding privacy over them to compute the total privacy loss or privacy budget. Luckily, all this came in handy in recent years, with the worldwide diffusion of deep neural networks, when differential privacy was applied to them. That is when databases became datasets and queries became (mostly training) iterations. We will see more in detail what is the result of this transition in the next paragraph §2.2.

2.3 Deep Learning with Differential Privacy

Differential privacy was initially devised for database analysis, but it is possible to apply it to deep learning models in several ways. In practice, since we primarily work with deep learning models, we will often refer to datasets rather than databases. However, this distinction does not entail any practical difference in the context of our discussion. Figure 2 shows how Differential Privacy can be incorporated in 1) the input data, 2) the gradients, 3) the output data, 4) the labels or even in the loss function 5). The most used technique is by training using DP-SGD (Differentially Private Stochastic Gradient Descent), while DP-Adam is less used - we discuss the reasons for this in §5.1.2. Both of these techniques involve gradient manipulation, falling under the point 2) in Figure 2. Initially presented by Song et al. (2013), DP-SGD is a variation of standard Stochastic Gradient Descent (SGD) that enhances privacy by incorporating gradient clipping and adding Gaussian noise. In DP-SGD, gradients are first clipped to a maximum norm to limit the influence of any single data point. Then, Gaussian noise is added to the average of these clipped gradients, ensuring that the updates to the model parameters maintain differential privacy by obscuring the contribution of individual data points. It is possible to consider the several layers separately in order to have different clipping thresholds, and the amount of noise should be calibrated over the sensitivity of the function (in this case, the network), as explained by Dwork et al. (2006b). The sensitivity of a function f indicates how much the function's output changes in response to changes in its input. In DP, there is also a focus on privacy costs, so DP-SGD needs a way to compute them. In Abadi et al. (2016), along with a refined version of DP-SGD, the authors present a novel technique for privacy accounting during training, which is called moments accountant. The composability property of DP allows to compute the privacy cost at every iteration and then sum them up. Privacy cost is computed in an asymptotic way, and while the mathematical details are beyond the scope of this survey, it is worth noting that moments accountant saves a non-negligible factor in the computation with respect to previous techniques, allowing to guarantee a lower privacy budget with the same magnitude of injected noise.

Also gradient clipping is an important step in the DP learning process. Gradient clipping can be done in several ways, it is possible to use a per-sample clipping (Lee & Kifer, 2020) and this can even be done automatically (Bu et al., 2022a), or a flat clipping style (also called all-layer) (Abadi et al., 2016) or something in between, like group-wise clipping (McMahan et al., 2017). As stated by Bu et al. (2023), the first leans towards better memory accuracy while the second favours memory efficiency. Another option to enforce DP is PATE (Private Aggregation of Teacher Ensembles) (Papernot et al., 2016) and its variations (Papernot et al., 2018; Tian et al., 2022; Boenisch et al., 2022). In this method, several teacher models are trained on

private data, and then a student model is trained on the predictions given by a majority vote with added noise made by the ensemble on public data. In this way, the actual model never sees the data, and the individual models (and, for extension, their data) cannot influence the outcome too much. A new, alternative way to enforce DP, as shown by Du et al. (2023b), is to perturb only forward pass embeddings, and according to the experimental evidence, it offers stronger performance with a lower vulnerability to attacks and more efficient memory usage. DP has been shown to work in several settings like NLU (Qu et al., 2021; Dupuy et al., 2021), image classification (De et al., 2022), diffusion models for image generation (Dockhorn et al., 2022; Ghalebikesabi et al., 2023), neural machine translation (Igamberdiev et al., 2023) and, of course, language modeling (Dinh & Fioretto, 2023) and LLMs in particular (Carranza et al., 2024). All these related works prove the interest of the scientific community in the problem and the commitment to finding an effective solution. Differentially private models are already in place with several solutions, and they work to different extents based on models, data and privacy budgets.

Federated Learning Federated learning (McMahan et al., 2023) is a machine learning approach that enables model training across decentralized devices or servers holding local data samples without exchanging them. Instead of centralizing all data in one location, the model is trained collaboratively across multiple devices or servers. Each local device computes updates to the model based on its local data, and only these updates are shared and aggregated to improve the global model. Federated learning aims to preserve privacy and security by keeping sensitive data localized while still benefiting from the collective knowledge gained during the collaborative training process. It allows to be involved in the training of a model without fully trusting the external entity training the model (which can be a government or a company), because you are not giving up your actual data. As we will see more in detail later, this technique does not really manage to protect privacy by itself because gradient updates, even after aggregation, can be reversed to recover training data §3.3.2. A big advantage of federated learning is allowing to train better models thanks to the higher availability of high-quality local data, like the case of next word prediction for smartphone keyboards (Hard et al., 2018).

Secure Multi-Party Computation Secure Multi-Party Computation (SMPC) (Yao, 1982; Zhao et al., 2019a) is a communication protocol that enables several parties to jointly compute a function over their private inputs, without revealing those inputs to each other. This way, each participant's data remains confidential while still allowing for the collective computation of the function. It is particularly useful in scenarios where data privacy is crucial, yet there is a need for collaborative computation, such as in privacy-sensitive data analysis and secure voting systems. There are several ways to enforce SMPC, like secret sharing (Shamir, 1979; Blakley, 1899), homomorphic encryption (Gentry, 2009; Kim & Yun, 2021; Joo & Yun, 2014), zero-knowledge proofs (Goldreich & Oren, 1994; Feige et al., 1988) and oblivious transfer (Rabin, 2005; Even et al., 1985; Brassard et al., 1986; Chailloux et al., 2013; Cheong et al., 2009). Homomorphic encryption is particularly interesting in this context as it allows the computation of a function on encrypted data. This makes it possible to perform training and inference of machine learning models on encrypted data. While these techniques can enforce strong data security, they also add a considerable computational overhead that makes them difficult to apply in situations where the learning process is already heavy.

3 Privacy Attacks

As the training data becomes huge, it is increasingly urgent to understand whether models tend to expose the data on which they have been trained since it is more likely for them to capture sensitive information. Moreover, the leakage of the original training data is not the only possible threat. In fact, the versatility of LLM is enhanced by the use of in-context demonstrations of the task that a user wants to perform (**?**). While being a useful technique, unconscious use can cause sensitive information – that should not be disclosed – to be included in these examples in context as well. For commercial systems, those data can also be leaked by exploiting vulnerabilities of the systems storing them (OpenAI, 2023). And even cyber security measures are not sufficient alone, since those data can be used in further training steps and subsequently leaked by a future version of the model. This aspect is also concerning for data with commercial value from a business perspective (Coles, 2023), and companies have decided, in some cases, to warn their employees about the responsible use of LLMs (Kim, 2023).

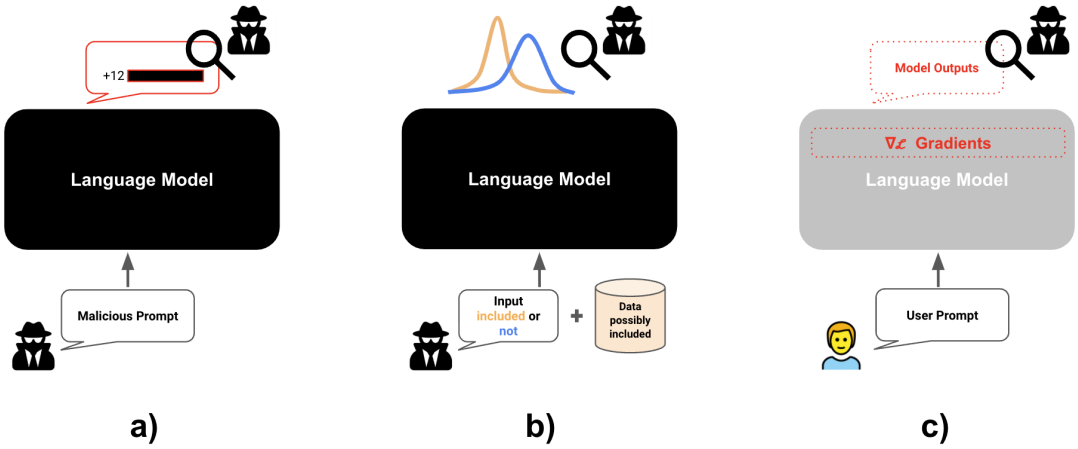

Figure 3: Attacks against LLMs aim to reconstruct private training data. **Training Data Extraction Attack (a):** The attacker has no internal access to the model (black box) and can only design a malicious prompt to force the model generation to reveal private and potentially sensitive training data. **Membership Inference Attacks (b):** The attacker has access to data that might have been included in the training set. The goal is to determine whether specific data points were indeed part of the training set. This is achieved by detecting behavioural differences in the model's responses to data that were included in the training set versus data that were not. **Model Inversion Attack (c):** In a black-box or grey-box setting, the attacker observes the model's outputs or gradients on unknown inputs. The attacker then uses this information to reconstruct the original input from these seemingly opaque sources.

A broad research area is devoted to understanding whether deep learning algorithms memorize training examples and, hence, can be used to reconstruct (potentially sensitive) training information. While some works question the effect of memorization during generation – meaning that texts generated starting from prompts extracted from training are different from the original ones (McCoy et al., 2023) – the discussion around memorization and its consequent risks is growing as larger models become more popular. In some cases, memorization is discussed as the key to good performance since data are often characterized by long-tailed distributions, hence, for a model, it is hard to distinguish between outliers and actually informative data points (Feldman & Zhang, 2020). However, memorization of training data by a model allows an attacker to reconstruct sensitive information or to understand whether a certain example was used during training. It has been shown by Carlini et al. (2023a) that it is possible to reconstruct training data with only black-box access.

For LLMs, it might be sufficient to input the right prompt, which is not necessarily an adversarial one (Carlini et al., 2021), in order to condition a model to generate sequences that are exactly memorized from the training data. Attacks that allow verbatim extraction of training data are generally referred to as Training Data Extraction attacks. When LLMs are fine-tuned with Reinforcement Learning with Human Feedback (RLHF), they tend to be more robust and avoid revealing the training set examples and, in particular, revealing sensitive information. However, some adversarial prompts have emerged to condition a model to show some unintended behaviour: since sensitive information is memorized by a model, RLHF is used to prevent the model from disclosing such information, but it does not ensure that new attacks cannot retrieve it. We discuss both non-adversarial and adversarial prompting in §3.1.

In some cases, an attacker may not only have access to a model but also to some data possibly included in the training phase of the target model. In this scenario, a Membership Inference Attack (MIA) can be performed: the attacker aims to claim that the information is included in the training set of a threatened model. The initial formulation of those attacks (Shokri et al., 2016) has also been extended against larger models and Transformer-based Language Models in particular. Since some sequences are memorized by a

model, overfitting over them can be detected and used to attest to the presence of certain data in the original training data. In §3.2 we discuss those kind of attacks.

When private data owned by different holders need to be used to train a joint model on them, protocols like federated learning (§2.3) seem to be the appropriate solution: in this scenario, data are not distributed across multiple servers, only model snapshots and updates are shared. However, also model outputs and gradients can be used to infer the input that justifies the observed model behavior. The model outputs or gradients can be inverted to obtain the sequence that justifies them, allowing an attacker to reconstruct private data without them being explicitly shared. We discuss Model Inversion Attacks as a method to track back model input from the model's output or gradients in §3.3). Moreover, from those rather opaque sources, like model outputs, an attacker can gain information regarding a model's parameters: those attacks, called Model Stealing attacks, while still not able to reconstruct entire LLMs, can potentially be directed against API-gated models, making them less opaque and, hence, more vulnerable against privacy attacks.

Finally, we notice that since fine-tuning is more expensive than in-context learning, many systems use in-context demonstration to specialize an LLM on specific tasks. Despite the flexibility of this method, some risks may emerge also by the usage of private information into in-context demonstrations: for example, a system that answer user query by leveraging a prompted LLM could be vulnerable to privacy attacks performed by a malicious user (see §3.4 for further details.)

3.1 Training Data Extraction

In Training Data Extraction attacks, an adversary aims to reconstruct training data having only access to a target Language Model. In particular, those attacks aim to extract the entire example by querying the target model (Carlini et al., 2021). Retrieval of examples is based on the assumption that they were memorized by the target models, and the definitions of memorization underlying these attacks may vary. However, the various definitions revolve around the key concept that a textual example or part of it can be extracted from a language model. In the original definition of Carlini et al. (2021), an example is defined to be extractable if there is a prefix used to condition the generation of the model so that the string can be exactly generated from the model. This definition is first introduced by Carlini et al. (2021) and then largely adopted with slight modifications (Huang et al., 2022; Carlini et al., 2023a; Nasr et al., 2023).

Training Data Extraction

Extracting Training Data consists of conditioning a LM to the right prompt so that it is likely that it will generate a training example to complete it. Formally, a textual example x is extractable from a LM g if there exists a prompt p such that: $x = \arg\max_{x'} g(x'|p)$. Once the right prompt has been identified, this procedure can be used to reconstruct training data, and hence, it can be used to extract sensitive information like Personally Identifiable Information (PII) from a model.

3.1.1 Non-adversarial extraction

Using training data extraction techniques, it has been shown that Language Models are prone to reveal private information when prompted with sentences that require the disclosure of such information. While trying to recover memorized training examples, Carlini et al. (2021) demonstrated that GPT-2 tends to memorize personal information such as Twitter handles, emails and UUIDs (Universal Unique Identifiers). In particular, in their experiments the language model generation is conditioned to some prompts that the attacker predefined: to maximize the probability of success of the attack, they choose prompts that are likely to be in the training data, like samples from the Common Crawl [2]. Then, once the generation is complete, the attacker tries to figure out which sequences are most likely to be generated as a result of the memorization, that is, which of the generated strings exactly matches the training data. In order to identify sequences that are more likely to be generated because of memorization, Carlini et al. (2021) compare the likelihood that is assigned to each sequence by two models: one is the model under attack – GTP-2 – and the other is a reference model, another LM. The attacker retains as possibly memorized sentences only those

[2] https://commoncrawl.org/

that have a high likelihood in the target model when compared to the likelihood assigned by the reference model. This is a form of Membership Inference Attack §3.2. Finally, once the attacker has retained a subset of strings that are likely to be generated due to memorization, the success of the attack must be verified by checking whether these high-likelihood sequences are actually part of the training set. However, for this step, Carlini et al. (2021) could only manually verify the effectiveness of the attack by searching the Internet for matching sequences, as the tested model did not have accessible training data.

The choice of the right prompt is crucial in this setting: in fact, when prompted with a prefix already seen during training, models tend to complete the prompt with the rest of the training sequence (Carlini et al., 2023a). Huang et al. (2022) further explore this direction and show that conditioning a model to a prompt already observed during training can lead to a leakage of PII, like emails, and that is more effective than designing new, previously unseen prompts. In particular, they distinguish between memorization and association. A model memorizes personal information if there exists a prompt – already part of the training data for that model – that leads the model to generate the personal information. The association requires that there exists a prompt – not necessarily observed during training and containing some reference to an individual – that can lead to the generation of PII by the model. Huang et al. (2022) show that the former approach is more effective than the latter, quantifying the number of e-mails that a model from the GPT-Neo family can predict when conditioned by the two different kinds of prompts.

Those attacks can also be used to reconstruct the dataset used to fine-tune the model. To reconstruct personal information from a generative language model like GPT-2, Lukas et al. (2023) propose to solve a masked language modeling task to reconstruct masked personal information from a sentence, conditioning a model on both the prefix and suffix of the masked information: they call this approach PII Reconstruction. However, it requires the attacker to gain access to masked training data before performing the attack. Panchendrarajan & Bhoi (2021) reconstruct the training data of a fine-tuned GPT-2. This work stems from the observation – shared also by Membership Inference Attacks as described in §3.2 – that sentences that have been used in the fine-tuning phase will be given different probabilities by the fine-tuned version of a model and by the base model. They generate candidate sentences by starting the generation of the fine-tuned model with an empty string. Then they observe the generation of the base model over the same sentence. Results show that it is possible to reconstruct about 75% of informative words in the fine-tuning dataset.

These experiments were the foundation for exploring other datasets and larger models. Nasr et al. (2023) demonstrate that the method proposed by Carlini et al. (2021) is more effective than initially thought: querying open-source models, they were able to verify the success of the attack procedure by accessing the training data only to verify the attack. They performed the attacks on models belonging to GPT-Neo, Pythia and RedPajama-INCITE families and using as initial prompts samples from Wikipedia. Wang et al. (2024a) explore the leakage of training data of OpenAI's GPT closed models evaluating the accuracy of information extraction of sensitive information contained in pre-training data. The private information they are able to extract is emails from the Enron Email dataset (Klimt & Yang, 2004). They hypothesize that this dataset is used during the training phase of GPT-3.5 and GPT-4 since it is included in the Pile dataset (Gao et al., 2020): while no information is available on the training set of those models, the Pile was used in the pre-training of the GPT-Neo family of models and hence, possibly, also in the closed models under investigation. They adopt the same prompts proposed by Huang et al. (2022) and show results also in few-shots prompting settings. The experiments demonstrate that GPT-3.5 and GPT-4 can predict, respectively, around 5% and 4% of the email addresses accurately. Moreover, in the few-shot prompting setting with emails extracted from the same dataset, these models accurately predict up to 48.19% of emails correctly. The same trend is not observed for prompts that contain emails that have a different domain from the target one, suggesting the importance of the right choice of demonstrations for training data extraction attacks. While LLMs are generally more resistant to attacks where association is required, the work of Shao et al. (2023) shows that it is possible to recover 3% of the training emails from a 20 billion parameter model with attacks that require association, meaning that the extraction procedure does not rely on prompts already observed during training. It has also been proposed to extract targeted examples from a model pre-training data: Zhang et al. (2023b) propose an attack to recover a certain suffix given a precise prefix, known to be in the pre-training data. They suppose that the attacker can also obtain some pairs of prefixes and suffixes and train an attack model on them: they tune soft prompt embeddings and then estimate the confidence

that a generated suffix is, in fact, as it is in the corresponding prompt in the pre-training data. A similar approach, based on soft prompt tuning, is also adopted by Ozdayi et al. (2023). As the size of the models increases, this type of attack appears to be more effective (Nasr et al., 2023). Moreover, the same techniques have been also discussed for other models, like diffusion models (Carlini et al., 2023b).

As concerns about the potential leakage of private information via training data extraction attacks grow, a number of tools are being investigated to assess and potentially prevent the leakage of private information by LMMs before it spreads. On the evaluation side, ProPILE (Kim et al., 2023) is a tool designed to test models with black-box and white-box attacks against their tendency to release PII: the black-box approach is designed to allow data subjects to test whether their PII is leaked; the white-box approach is based on soft prompting and allows providers to conduct their model surveys to prevent leaks of private information. Differential privacy (see §5 for further details) is also effective against the memorization of private information and can successfully defend against these attacks. Downey et al. (2022) demonstrate that by hand-crafting canary sentences containing sensitive content in the training data: the last two words of each canary are designed to contain the most sensitive information and the LM is asked to complete the canary. The GPT model they train with Differential Privacy – with some loss in model performance – successfully preserves privacy.

3.1.2 Adversarial prompting

There have been initiatives to ensure LLMs are aligned with human values and used appropriately. For example, the OpenAI usage policy requires that users do not use the models to compromise the privacy of others, including collecting or generating personal data. Moreover, OpenAI's ChatGPT has been aligned with Reinforcement Learning from Human Feedback (Ouyang et al., 2022) and avoids answering questions regarding its training data. In this scenario, the techniques based on non-adversarial prompts may not be effective: Nasr et al. (2023) discuss how ChatGPT tends to not expose training data when prompted with prefixes that are likely seen during training. For these reasons, some attacks against LLMs define adversarial prompts that force the model to diverge from its originally intended use. For example, asking the model to repeat indefinitely a word, Nasr et al. (2023) notice that ChatGPT diverges and starts emitting training data, also containing personally identifiable information.

A new form of adversarial prompt attacks are jailbreak prompts. These prompts are designed to circumvent safeguards and manipulate LLMs into generating harmful content. In some cases, a goal hijacking is performed, meaning that the original goal of a prompt is changed to a new goal, like generating a target phrase, by injecting malicious text in the original prompt (Perez & Ribeiro, 2022). Shen et al. (2023) characterize a large set of jailbreak prompts and evaluate their effectiveness against different LLMs. They notice that most of these prompts encourage the model to act like another fictional assistant, such as DAN (short for Doing Anything Now), which has no limitation in the answers. Along with other potentially harmful scenarios, they show that this type of prompt allows a user to receive suggestions on how to collect personal information.

In this context, Li et al. (2023b) focused on the usage of adversarial prompts to pose privacy threats: they propose a multi-step jailbreak prompt to extract personal information, such as emails, based on Chain-of-Thought (Kojima et al., 2023). They study the effect of adversarial prompting against ChatGPT. They first notice that the model tends to avoid sharing private information if direct data extraction prompts are used. Then they propose the Multi-step Jailbreaking Prompt in three steps: they assume the role of the user to input the jailbreaking prompt, then act as the assistant to confirm that the jailbreak mode is enabled and, lastly, they play the role of the user and query the assistant with a non-adversarial prompt. They show that more frequent emails are less difficult to recover: experiments on the Enron dataset show that more than 50% of frequent emails in the dataset can be extracted.

3.2 Membership Inference Attacks (MIA)

Membership inference attacks (MIA) (Shokri et al., 2016) are currently a de facto standard when it comes to verifying model privacy (Murakonda & Shokri, 2020). In this kind of attack, the attacker has access to (1) **some records** (supposed to come from a similar distribution of the training data) and (2) to **the model**, which could be provided as an API, in a black box setting, or be fully available as a white box. The aim

of the attacker is to find out if these records (1) are actually part of the training set for that model (2). The initial formulation of this attack by Shokri et al. (2016) required the attacker to build several "shadow models" that mimic the original model, trained on datasets that may resemble the distribution of the original one. Then, the attacker trains a classifier on the shadow models' outputs to assess the usage of some records during the training phase by the shadow models (where the ground truth is known since the attacker also designed the training set of those models). The classifier, trained on shadow models' outputs, is then used against the original model with the objective of identifying data that are part of its training set.

MIA as Classification of Behavioural Changes

MIA can be framed as a classification problem: the attacker determines whether a certain example has been used during the training phase of the model or not. Attacks are based on detecting changes in the target model when it is exposed to data points that are in the training set versus when it is exposed to examples that are not part of it.

Stemming from the general setting described in Shokri et al. (2016), MIAs have been successfully tailored for Language Models to extract sensitive information from them (Song & Shmatikov, 2019; Hisamoto et al., 2020). Carlini et al. (2022) observe that MIAs could be better framed as the procedure of comparing models that are trained on a given example and models that are not trained on it, and to find a difference in their behavior, for example having a likelihood of the example much higher in the first case rather than in the other. They train numerous shadow GPT-2 models to measure the probability of observing a certain likelihood of an example in models trained and not trained on it.

However, training shadow models may be too expensive in the context of LLMs: the original formulation of MIA can be used as long as it is feasible for the attacker to construct shadow models that are similar to the target model.

3.2.1 MIA with Thresholds

Since the construction of shadow models can be too expensive, MIAs have also been formulated without relying on them. In fact, some MIAs have been designed to exploit some measure of overfit over the training data in order to understand whether a given example has already been seen by a model: overfitting the training data is a sufficient condition to detect memorized examples (Yeom et al., 2018). Shi et al. (2023) observe that the probability distribution is more spread over low probability words for an unseen example: they design attacks that threshold the average log-likelihood of the top-k rarest words to ascertain whether or not the example is part of the training data. Language models trained on clinical data have been shown to be sensitive to MIAs. Mireshghallah et al. (2022) demonstrate the effectiveness of MIAs designed against Language Models trained with Masked Language Modeling objectives, obtaining a substantial gain both in AUC and in recall of the attack over the baseline. Their method is based on thresholding the ratio between the likelihood of a sentence on the target model and the likelihood of the same sentence on a model trained on a general population. Mattern et al. (2023) argue that the limited availability of training data for the reference model poses a strong restriction on the applicability of previous work. They propose to generate perturbations of a sentence and to measure the average loss of a target model on the original sentence and on the perturbations: they hypothesize that, if the sentence is not in the training set, the model should behave similarly also on the perturbed examples, while it should have a lower loss conditioned on the original sentence if it is part of the training set. These kinds of attacks are also useful to understand, given two snapshots of a model – for example a pre-trained model and a fine-tuned version of it – on which data the fine-tuning phase was performed. In particular, the difference between the probability assigned from different snapshots to each token in a sentence can help identify whether an example was used during later training phases (Zanella-Béguelin et al., 2020). Duan et al. (2023) show that LLMs, while being extremely versatile in a number of tasks thanks to their in-context capabilities, can be subject to MIAs that target datasets used during prompt training. They demonstrate that an LLM trained to answer to prompts tends to expose more information than the corresponding fine-tuned model.

However, a recent body of research poses some critical challenges to the use of MIAs to assess their reliability in demonstrating the inclusion of a sample in a LLM training data. In some cases, the theoretical foundation

of MIA is discussed. Zhang et al. (2024c) argue that, since it is no longer possible to use shadow models that do not include the data point in their training set, the statistical framework behind MIA is no longer applicable since other approaches are statistically unsound. Duan et al. (2024b) demonstrate empirically that a range of MIA are not reliable against LLMs. According to their findings, training for a single epoch – as is becoming standard practice for LLMs – as well as increasing amounts of training data may negatively affect MIA accuracy. They also notice that MIA become more difficult as the difference between data included in the training set and data not included becomes more subtle. However, as training corpora become larger – their statistics are based on the Pile (Gao et al., 2020) – this similarity increases: for example, non-members Github samples share on average the 76.9 % of their 7-grams with the Pile. Hence, the combined effect of training techniques that diverge from the one defined for other models and the huge training sets may challenge MIA effectiveness against LLMs.

3.3 Model Inversion and Stealing

Model inversion attacks generally exploit the output of the model and use this information to reconstruct the input data that generated the observed outputs. In the initial formulation (Fredrikson et al., 2014; 2015), an adversary is given access to the trained model and to a partially masked input, for which only non-sensitive features are disclosed. The attacker aims to infer the value of the sensitive attributes that serve as input to the model. While the initial formulation is implemented for linear regression models (Fredrikson et al., 2014) and neural network trained for image processing (Fredrikson et al., 2015), this type of attack has been successfully used to extract sensitive input information from different models, including LSTMs (Welleck et al., 2018). Those kind of attacks are challenging against LMs especially since, in principle, the optimization process to reconstruct the initial input of a model have to deal with the discrete nature of text (Parikh et al., 2022). Model inversion attacks are often performed against distributed models: in fact, some protocols like federated learning ensure that private data are not shared across multiple servers (see §2.3 for further details). However, with some access to model outputs, it is possible for an attacker to reconstruct the input data that causes the observed output.

Model Inversion Attacks

This family of attacks threatens models especially when they are distributed across multiple servers: in fact, while data holders do not directly share private data – which are thus maintained locally– they may be willing to share seemingly opaque information related to a machine learning model, such as model outputs, embeddings or gradients. However, Model Inversion attacks are designed to reconstruct private data from these apparently opaque sources.

In some cases, from those opaque sources, it's also possible to reconstruct a model's parameters from its output: in this case, the attack technique is often defined as Model Stealing (§3.3.3). These attacks aim to recover model weights from an API-gated model, making it less opaque and, hence, less robust also against other type of privacy attacks. While the stealing of entire LLMs is not described yet, some works efficiently reconstruct a subset of the model parameters in black-box attacks.

3.3.1 Model Output Inversion

The outputs of a model, like the representation of textual data that a Transformer-based LM produces in the last layer, can be totally or partially inverted to reconstruct the text that is originally the input of the model. In particular, these representations – often called embeddings – are only apparently opaque and can be inverted to reconstruct the entire sentence or the words that it contains. Pan et al. (2020) design the first model inversion attack against Transformer-based Language Models. They observe that directly framing the inversion problem as an optimization problem is challenging due to the discrete nature of sentences. They train an attack classifier that is designed to extract fixed patterns and keywords – like, for example, location or medical information – from sentence representations of models. Similarly, Song & Raghunathan (2020) predict unordered sets of words from sentence embeddings: their attack model is a neural network trained with the multiset prediction loss.

In some cases, the entire input text is also reconstructed. Li et al. (2023c) implement the inversion as a generative language modeling task with a decoder-only model, conditioned on the embedding of the sentence to invert as the first token representation. They show that the model successfully inverts meaningful tokens containing a large fraction of the named entity. Morris et al. (2023a) propose an iterative method to reconstruct the input text of Transformer-based models that produce embeddings for documents: their attack is based on the idea of iteratively generating different hypotheses that may justify the observed embeddings. They learn the distribution of texts in a dataset given their embeddings: their approach starts by assuming an initial hypothesis and then iteratively improves it by adjusting the hypothesis to progressively align its embedding to the target one. This approach has been extended also to a multilingual setting, adding a translation step before iterating again over the generated hypothesis Chen et al. (2024). Only observing the logits of LLMs can be also sufficient to reconstruct the original input: Morris et al. (2023b) train an inversion model on a collection of instruction datasets to reconstruct the prompt of an LLM given the probability distribution of the token following the prompt. In a completely black box scenario – in which only model's output is observed – an attacker may be able to reconstruct the original prompt training an inverter – an other LM – on previously observed outputs on known input prompts Zhang et al. (2024a).

This kind of attack has also been applied to Language Models fine-tuned on classification tasks to reconstruct private training data. In this case, the output of the model is a probability distribution over each of the possible classes. This information can be effectively exploited: in particular, an attacker may exploit the greater confidence that a model has in classifying sentences that are already part of the training set. Elmahdy et al. (2022) inject canaries in the training data (sequences that are intended to be later extracted and out of distribution with respect to the other data) and then reconstruct a partially masked sentence to search for tokens that maximize the probability of the target label. They perform the reconstruction by enumerating tokens in the vocabulary, weighted by the number of occurrences that they have in the training data. Zhang et al. (2022c) propose instead to build a dataset that resembles the unknown training data and initially train a text generator on these collected data, similarly to what is done with shadow model MIA (§3.2). Then, they further train the text generator via word embedding perturbation with the objective of minimizing the classification loss on the target model. The idea is that the target model's prediction on training sentences will exhibit a lower loss and this information can be used to perturb the text generator model and obtain data closer to the private one.

3.3.2 Gradient Inversion

In federated learning (Ray et al., 2021) multiple nodes train a model cooperatively without directly sharing data samples: the system is designed to make training feasible while exchanging model updates rather than data across nodes (see §2.3 for further details). Gradient Inversion attacks are directed against federated learning environments that perform training by sharing gradients. Those attacks can be used to reconstruct training data from seemingly opaque sources.

Zhu et al. (2019) demonstrated that if the gradients are openly accessible, it is possible to reconstruct the training data. The attacker aims to reconstruct the training data by minimizing the distance between some generated gradients and the ground truth one. In particular, the attacker initially generates gradients from random inputs and labels. Then, having access to the ground truth gradients solves the optimization problem of minimizing the L_2 distance between the generated gradients and the ground truth one, parameterized by real inputs and labels. In their experiments, Zhu et al. (2019) applied this attack against a BERT model. This idea was further explored by inverting gradients for image classification tasks (Zhu & Blaschko, 2020; Geiping et al., 2020; Yin et al., 2021).

In reconstructing textual data, Deng et al. (2021) further attempted to perform a gradient attack against Transformer-based language models introducing also a L_1 term in the reconstruction loss. Experiments on a Transformer model, BERT model and TinyBERT demonstrate the effectiveness of this gradient attack, recovering up to 50% of the original tokens on average. In LAMP (Balunovic et al., 2022), more effective results are achieved by simultaneously training the attack model to minimize the difference between the reconstruction gradients and choosing at each iteration only sequences that have low perplexity according to an external language model (like GPT-2). Reconstructing the correct word order is, in fact, one of the

main challenges: Gupta et al. (2022) propose to recover from the gradients a bag of words for the sentence to extract and then perform a beam search to effectively reconstruct the sentence.

3.3.3 Model Stealing

As exposing only black-box access to a LLM is a standard practice for popular, commercial models, an adversary might want to reconstruct the gated model with the aim of imitating its behaviour. A family of attacks called model stealing attacks have been designed with this goal. The adversary, in this case, can have no information regarding the original training data. The attack usually requires observing the model outputs on inputs that the attacker fed to the gated model to reconstruct its internal parameters. Tramèr et al. (2016) introduce this family of attacks and demonstrate that an attack could lead to a accurate reproduction of logistic regression, neural networks, and decision trees. They also successfully demonstrate on those systems that model inversion as defined by Fredrikson et al. (2015) could be applied to the stolen model to reconstruct private input, with direct privacy implications.

Krishna et al. (2019) demonstrate the feasibility of those attacks against Transformer-based models. They use random words as input for gated fine-tuned BERT models and use their outputs for fine-tuning an equivalent model. They demonstrate also that the quality of the stolen model increases if the attacker has some background knowledge regarding the domain of the original fine-tuning dataset. Moreover, Zanella-Beguelin et al. (2021) notice that an attack that exploits the characteristics of the Transformer last layer can effectively reconstruct a fine-tuned BERT model with frozen pretrained weights and learnable embedding projection matrix. In particular, they reconstruct the embedding projection matrix by feeding the gated model random inputs and collecting the corresponding output. They reconstruct the embedding projection matrix optimizing its weight on the observed embedding and outputs. Their technique, in combination with other techniques like the one proposed by Krishna et al. (2019), can lead to a refined model, that resembles the target one.

Recently, a similar idea has been applied to threat LLMs: Carlini et al. (2024) proposed an attack to recover the embedding projection matrix at the last layer of a transformer language model. They attack a model in black-box, collecting model's output logits on random prompts. They notice that since the embedding projection matrix is low rank – since it projects the hidden dimension to a logit vector that has higher dimensionality – it's possible to apply the SVD algorithm to the collected logits to obtain a linear transformation of the original embedding projection matrix. Moreover, they demonstrate that the logits over the entire vocabulary are not necessary, and that the probabilities over a predetermined number of words are sufficient for a successful attack. A similar approach has been also proposed by Finlayson et al. (2024) and they describe the utility of the attack in discovering also whether a gated model has undergone some update.

3.4 Privacy Threats at Inference Time.

In the LLM context, the leakage of the original training data is not the only possible threat. The inclusion of additional, potentially private information at inference time presents a number of challenges. Here we summarize three of the major challenges derived from such systems. These threats have pushed a great deal of research to focus on how to perform calculations securely on sensitive user input, either by obscuring them through differential privacy (§5.2) or by keeping them encrypted during the entire calculation process (§5.5).

Computation over Private Input. Given the challenges of training such models from scratch, the option of using LLMs in a Machine Learning as a Service modality is becoming increasingly popular. In this setup, an organization hosts the model and users simply submit their data for processing. This approach raises significant privacy concerns for user data: vulnerabilities in the systems storing this data could lead to breaches (OpenAI, 2023), and model providers must offer strong assurances to comply with privacy regulations, such as GDPR. Similar concerns even led to a temporary ban on ChatGPT in Italy (Lomas, 2023). This issue is particularly concerning for businesses dealing with proprietary data or commercially valuable information (Mitchell, 2023; Coles, 2023), and companies have decided, in some cases, to warn their employees about the responsible use of LLMs (Kim, 2023).

Prompting Private Data. At inference time, the versatility of LLM is enhanced by the use of additional data, which are often provided in the form of demonstrations in the context of the task the user wants to perform (Brown et al., 2020). While being a useful technique, some sensitive information could be included in these examples in context as well and an increasing body of research suggests that some risks can be identified during inference. In particular, since in-context learning is more cost-effective than fine-tuning, some systems may deploy LLMs that handle user requests by utilizing pre-loaded in-context demonstrations, which may include private or sensitive information. This efficacy, however, comes with an increased risk in terms of privacy leaks, which can also outweigh the risks against fine-tuned models (Duan et al., 2023). Wu et al. (2023a) propose a threat model in which a user of a system submits some data containing private information, such as medical records, and another user of the same system could potentially access that sensitive data. They focus on addressing this threat through the application of Differential Privacy during inference (see §5.2 for further details). Duan et al. (2024a) demonstrate the efficacy in a membership inference attack (MIA) against prompts containing sensitive, in-context learning demonstrations: they suppose that a model is prompted for a classification task and that the adversary may want to understand whether a certain example has been used or not as in context demonstration. They argue that an adversary can test whether an example has been used in context since the probability for the corresponding class will be much higher for members (in context examples) than for non-members. Similarly, in this scenario, a malicious user –instead of performing the intended task on the system– can potentially extract the in-context demonstrations via jailbreaking attacks (Tang et al., 2024). Chu et al. (2024) demonstrate that entire conversations can be reconstructed via jailbreaking when they are used as contextual information for prompting an LLM.

Contextual Privacy. As LMs' capabilities increase and their role as assistants becomes more prominent, it is important to understand whether they are able to distinguish if a piece of certain sensitive information is safe to be shared in a given context. For example, there is no privacy leakage when a doctor shares with his patients information about their health. However, the disclosure of such information to an unauthorized insurance company violates patients' privacy. Those considerations are recently emerging as language models can be reliably used as assistants by groups of users that want to share some information while preserving others. Brown et al. (2022b) discuss the discrepancy between data sanitization and differential privacy that assume a structure of private data and cannot solve the necessity to share private information only to the correct group of people. They also argue that some data protection techniques – like data sanitization and differential privacy – make assumption on the nature of the data that should be protected: implicitly or explicitly they state that a private information is structured information, that should not be shared. However, every bit of language can potentially contain secrets that are not meant to be shared indiscriminately. Mireshghallah et al. (2023) formalize the necessity to share secrets in the appropriate context stemming from the contextual integrity theory (Nissenbaum, 2004). In particular, the information to be protected is not only the one in the training data but also during inference. They introduce a new benchmark to assess the ability of such models to disclose private information only to users who should be allowed to access them. They measure that multiple LLMs include information that should not be disclosed when asked to generate summaries of meeting conversations. Priyanshu et al. (2023) similarly explore the tendency of ChatGPT to disclose personal identifiable information when asked to produce summaries of hiring decisions and medical notes, a behavior that they notice can be mitigated by explicitly asking the model to comply with privacy regulations and feeding step-by-step prompts that suggest which information should be deleted. Wang et al. (2024a) also discuss the necessity to measure the privacy leakages during inference. They focus on PII like emails and phone numbers: they propose to initially feed the model with prompts that contain PII with the indication that those should not be shared; then, they prompt the model asking for the same information. They notice that GPT-3.5 is prone to violating the privacy restriction, while GPT-4 is more robust. Both models, however, tend to reveal private information when prompted with some demonstrations containing leakages in the few-shot scenario.

4 Solutions for Preserving Privacy in LLMs: Data

After identifying the key vulnerabilities and potential attacks on LLMs as outlined in §3, in this section, we introduce the reader to the current possible solutions to the problem of preserving privacy in the context of LLMs and text generative models. We have to say that the field is very fragmented as this is a very

recent concern and the field is evolving fast. There is not, at the moment of writing this survey, a single gold standard solution to the problem of preserving privacy in LLMs and text generative models. The main solutions available can be broadly categorized into two categories: **i) Preserving Privacy in Data** §4 and **ii) Preserving Privacy in Model** §5. In the former section, we collected all the methods that try to solve the problem at its source i.e., they act directly on the privacy-sensitive information in the dataset, before the training happens, while in the latter we selected all the methods that address the problem from the model perspective i.e., try to remove from the model problematic information after the training. The two approaches fit different use cases, so it is important to be aware of both.

In this section, we propose several solutions that allow preserving privacy from the *data perspective* i.e., we preserve privacy directly in the training data used in machine learning models. This includes any approach that involves manipulating, carefully choosing, or generating training data in order to avoid the memorization and subsequent leakage of private information inside a large language model or any model that learns from data. This basically puts the focus on preserving privacy in the *data* used to train the model. Given that a machine learning model is molded by its training dataset, training it on a privacy-preserving dataset ensures that these methods offer equivalent privacy assurances to the dataset itself. The methods presented here are thus not limited to LLMs as they can be used to anonymize textual data more generally. That is, it is possible to use them to strip datasets of sensitive information before using these datasets to train LLMs. Making the training data safe means making the trained model safe.

In the context of LLMs, there are very few technical reports that thoroughly describe the precautions necessary to preserve privacy via data anonymization, and even fewer that release the preprocessing software used for this anonymization. Among the notable examples, the LLaMA3 technical report (Dubey et al., 2024) mentions "filters designed to remove data from websites [to eliminate] unsafe content or high volumes of PII" but does not detail the exact procedure, while noting the use of "Microsoft Presidio Analyzer" in the preprocessing of speech data (See our overview in §6). Similarly, Soldaini et al. (2024), in the context of training OLMo (Groeneveld et al., 2024), also considered using Microsoft Presidio but ultimately opted for custom regular expressions, given the scale of the pretraining dataset.

Overall, the available open reports suggest that for removing PII from the *pretraining dataset*, the preferred solution at scale tends to involve custom regular expressions (Penedo et al., 2024). The broader subject of preserving privacy in *pretraining data*, however, remains largely unexplored beyond these rudimentary techniques (Subramani et al., 2023). The scope of this section is thus two-fold: i) propose to the reader possible new avenues to preserve privacy in the *pretraining* dataset by surveying existing approaches proposed in the literature that have not been applied yet in the context of the *pretraining phase* of LLMs; ii) propose to the reader anonymization techniques that can be used to anonymize data for the *instruction-tuning / fine-tuning* phase where the LLMs are possibly fine-tuned on sensitive in-domain data for the downstream task (e.g., healthcare).

The section is structured in the following way: we first show classical anonymization techniques §4.1, then we focus on anonymization techniques built with Differential Privacy §4.1.2. Putting this last section in relation with Figure 2, the point of application of DP would be (1), the input, because all these techniques aim to apply Differential Privacy on the data before feeding it to the LLM (for either training or inference).

Preserving Privacy in Data

Key Idea: Preserve privacy directly on the dataset used to train the model.

4.1 Anonymization

Data anonymization is not a new concept, with seminal work dating back to the 1990s: a pivotal development was the introduction of k-anonymity in 1997 (Sweeney, 1997), which means ensuring that data cannot be re-identified to fewer than k individuals. This foundational concept has spurred extensive research, leading to a plethora of anonymization techniques. Given the breadth and depth of the field, this section is organized into two main parts: the first, titled Classical Anonymization 4.1.1, explores traditional methods such as rule-based approaches and includes a discussion of how recent advancements in language models

Figure 4: **Classical Anonymization Pipeline.** The figure illustrates the standard process for anonymizing data in natural language processing tasks. The pipeline consists of three sequential blocks: Preprocessing, Identification, and Anonymization. The Preprocessing block prepares the raw data for further analysis. The Identification block involves the use of a procedure to detect potential Personally Identifiable Information (PII) within the data (e.g., Named Entity Recognition). Finally, the Anonymization block replaces identified PII with anonymous substitutes, ensuring privacy preservation in the data. Arrows indicate the flow of data through each stage of the process.

have been leveraged for anonymization purposes. These traditional methods have been the cornerstone of anonymization for decades. In more recent developments, the rapid adoption of Differential Privacy has marked a significant evolution in the field, offering robust privacy guarantees that, depending on the application context, can surpass those of classical methods. The second part of this section, Anonymization with Differential Privacy (§4.1.2), discusses these techniques. It also introduces the concept of Metric Differential Privacy, highlighting its unique applications and benefits within the field of LLMs and beyond.

4.1.1 Classical Anonymization

As already mentioned, anonymity definitions can be traced back to 1997 with k-anonymity. Anonymization aims to make impossible the re-identification of an individual, that is the process of reconstructing the identity of an individual given a record that contains their information. Different attributes could be anonymized. While effectively masking PII – like names, emails, and social security numbers – is necessary, some features that are not exactly identifiers can be used in combination to identify an individual: those features are called quasi-identifiers. Quasi-identifiers are apparently non-personal attributes – like gender, date of birth, and zip codes – that can be used in combination with external resources to identify a person from its anonymized records. The definition of k-anonymity (Sweeney, 1997) ensures that quasi-identifier cannot be used for re-identification. Essentially, a dataset is said to have k-anonymity if any given record in the dataset is indistinguishable from at least k - 1 other records with respect to certain identifying attributes called quasi-identifiers. Upon this concept, several implementation techniques were proposed by Samarati & Sweeney (1998), like generalization (modifying attribute values to less specific but accurate representations), suppression (i.e., entirely removing data that are outliers or too identifiable), minimal Generalization (ensuring data are not generalized more than necessary to meet k-anonymity). Of course, this method is highly susceptible to the choice of k (even inapplicable sometimes) and gives coarse data, making it impossible to carry out fine-grained analysis. It is also important to mention that, due to the reliance on quasi-identifiers, this method is much less effective with unstructured text. Since anonymization has such a long history – with the first paper theorizing k-anonymity dating back to 1997 (Sweeney, 1997) – much research was devoted to this field, resulting in the development of several techniques. Larbi et al. (2022) give an overview of many anonymization techniques, with their different pros and cons based on the use case. In this section, we will show the most relevant ones that are also available in a synthetic tabular format in Table 2.

The standard process is outlined in Figure 4. After any necessary preprocessing phase, the first challenge to solve is to identify the entities that may cause a violation of privacy. This can be done via a Named Entity Recognition (NER) tool. To maintain the coherence of the text, a Co-reference Resolution (CRR) system can be used to resolve references between entities should be identified if the anonymization framework aims to ensure consistent anonymization across the document. Finally, private information should be replaced with anonymous information. The anonymization methods can vary. Suppression is used to completely remove the information. Tagging allows the replacement of sensitive information with artificial labels –like the one obtained by the NER system– that still retain information about the class of the data and an identifier. The sensitive information can be also replaced with other textual information: random substitution and

generalization can both be used. Random substitution uses another random entity of the same class, while generalization replaces an entity with a more general term.

Anonymization before DNNs Mamede et al. (2016) present a modular anonymization framework for text documents in Portuguese that closely resembles the general workflow for this task. The system consists of four modules: pre-processing, Named Entity Recognition (NER), Co-reference Resolution (CRR), and Anonymization. They notice that the effectiveness of automatic text anonymization methods relies heavily on the performance of Named Entity Recognition (NER) and Co-reference Resolution (CRR) modules and while CRR shows good performance, the NER can be very lacking depending on the dataset. They also implement and test a number of different anonymization strategies: all the anonymization methods presented can be effective in some cases, but they all exhibit some drawbacks. Suppression, while dependent on NER accuracy, often compromises text comprehension due to loss of information, challenging readers to grasp entity references. Tagging helps maintain references between anonymized entities but detracts from natural text flow, necessitating improved NER and CRR accuracy. Random substitution maintains natural language output but can lead to semantic drifts due to arbitrary replacements, suggesting a need for a curated list of vague yet contextually appropriate entities. Generalization offers a balanced approach by replacing specific entities with broader categories and improving readability and relevance, especially for location entities, though it struggles with accurately linking entities with multiple knowledge base entries.

Another prominent anonymization technique is perturbation, as proposed by Zuo et al. (2021). It involves subtly altering the original data to preserve statistical significance, aimed at reducing the vulnerability of the dataset to linkage and enhancing privacy protection. Techniques like microaggregation and data swapping are commonly used for the implementation of perturbation. Microaggregation involves grouping data and using representative values, while data swapping involves interchanging values between records to maintain the overall statistical properties of the dataset. While perturbation effectively anonymizes raw data and retains its statistical utility, it may compromise the accuracy of anonymized data.

Anonymization with DNNs More recent anonymization methods also leverage Deep Neural Networks. Zhou et al. (2022) use BERT to impute previously masked sensitive information. This framework is called DataSifterText and it works with a whitelist that holds sensitive terms that need to be masked and a blacklist for meaningful terms that should be retained. After a proportion parameter is chosen, sensitive tokens are masked according to that proportion, and then BERT is used to replace the masked tokens. These two steps in the procedure respectively cover the identification and anonymization step shown in 4). Results show that even when the obfuscation level is high, the framework retains significant data utility, demonstrating the method's effectiveness in maintaining informative content.

In some cases, the author of a text is meant to be identified: re-identification of an author can be done by spotting some features that characterize an author's writing. Romanov & Fedotova (2020) analyze this scenario for the Russian language. They extract features like lemmas, parts of speech and morphological features. Then, a fast correlation filter – a multivariate feature selection method where the class relevance and the dependency between each feature pair are taken into account – is used to select the most informative features. Those informative features are the more likely to be identifying features. To preserve the privacy of the author, those features are smoothed by making targeted adjustments: for example, words or punctuation marks that are characteristic of the author's style are replaced or altered based on average statistics derived from a broader corpus. The final step is to further process the smoothed text with a deep learning model: in this work, a Transformer-based model is chosen. This approach reduces the probability of authorship identification to random guessing.

Mosallanezhad et al. (2019) use an attention-based task-aware text representation learner for extracting latent embedding representations with a reinforcement-learning-based privacy and utility preserver for manipulating these embeddings. While it effectively balances privacy and data utility and is adaptable and efficient, it has notable drawbacks, including complexity, potential data-specific limitations, and dependency on precise parameter tuning.

More recently, some works proposed the use of LLMs for anonymization. Vats et al. (2023) propose to hide sensitive tokens via a masking process and impute them by using a Large Language model (LLM). In relation

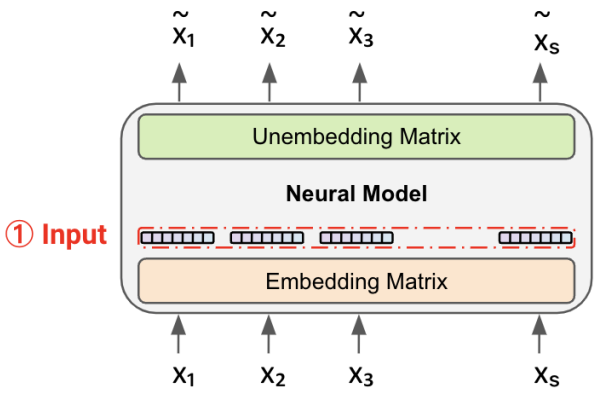

Figure 5: **Anonymization with Differential Privacy.** This figure illustrates the procedure to anonymize data via differential privacy. A neural model is used to embed data, data embeddings are then perturbed with noise using DP. The anonymized output is then returned by decoding the perturbed embeddings.

to Figure 4 the masking acts as the identification part, as it finds and masks sensitive information, while the imputation represents the proper anonymization, where non-sensitive terms replace the (previously masked) sensitive ones. To carry out the masking process, several techniques are proposed: a handcrafted list of safe words (the ones that do not carry sensitive information) called *allowlist* created by linguistic experts, this is similar to the blacklist of Zhou et al. (2022) and identifies the terms that do not need to be masked; another method to identify safe words is vocabThres, where the N more frequent words in a vocabulary are considered safe; the last masking criteria is to find the unsafe words and involves using a NER module that identify potentially private tokens to mask them (similar to Mamede et al. (2016)). After the choice of a masking method, the chosen tokens are masked and then imputed by an LLM. Also for the imputation, three separate methods are proposed: Top-1, which uses the one token with the highest probability from the LLM's output; Top-K, which randomly selects one of the first K tokens with the highest output probabilities; Fine-Tuning, which selects the records that do not contain sensitive information and uses them to fine-tune the LLM, then just takes the highest probability output (so after fine-tuning it circles back to the Top-1 approach). They compare models trained on the anonymized datasets against models trained on the masked dataset, showing a significant improvement in both tasks.

As the usage of closed models increases, privacy concerns also emerge when sending prompts containing private information. Chen et al. (2023b) focus on sanitizing prompts before sending them out to some unsafe third party like a black-box model like ChatGPT, thus they enforce prompt privacy. Moreover, they argue that standard anonymization techniques – like masking – may be useful to ensure the privacy of prompts, but the model output could not be effectively de-anonymized to improve its utility. They propose a new framework called HaS (Hide and Seek) with two local small LMs, one trained for anonymization and one trained for deanonymization. The first one preprocesses the prompt before sending it to the black-box and the second one postprocesses the model output. The hide model is trained with original sentences and extracted privacy entities and is prompted to randomly substitute them, while the seek model is trained with the same data and the addition of the remote LLM answer. The framework is evaluated using both adversarial models (black box and white box) to measure privacy protection effectiveness and usability experiments to assess the impact on LLM task performance, including translation and classification. Adversarial models are unable to recover original information, indicating stronger privacy protection. Usability is evaluated by metrics like BLEU for translation and precision, recall, and F1 scores for classification, to see how well the framework maintains the utility of LLM outputs post-anonymization. The results demonstrate that HaS effectively balances privacy and utility, with label-based anonymization showing better privacy protection and generative methods offering slightly better utility retention.

4.1.2 Anonymization with Differential Privacy

In this section, we are proposing works that make use of Differential Privacy for data anonymization i.e., before actually using the data in training. We refer the reader to §2.2 for an introduction to Differential Privacy, and we will propose later in §5 methods that use Differential Privacy directly on the model. The methods proposed in this section apply Differential Privacy to perform text rewriting, where the original text is modified by perturbing its representations to ensure privacy. The key advantage of this approach is that it adheres to the formal guarantees of Differential Privacy. All the works in this section are also available in tabular format in Table 3.

Referring to Figure 5, we identify the works in this section as examples of Differential Privacy applied to the input (1) of a neural model. Differential Privacy can also be applied in a separate, typically smaller, auxiliary preprocessing model to anonymize data before it is fed into a LLM.

Metric Differential Privacy Feyisetan et al. (2019b) propose a DP mechanism to transform text data that constructs a hierarchical representation to identify private phrases in the input sentence and then randomly replace them with neighbouring words in a word embedding space. They build upon Metric Differential Privacy that, as explained by Chatzikokolakis et al. (2013), extend Differential Privacy principles to location-based privacy. This approach is adapted for textual data by utilizing word embeddings as coordinates, transitioning its use from spatial to textual domains. For obfuscation, target words are selected by favouring locations near the original one with a higher probability over distant ones. For a graphic representation of this process refer to 5. Building further on the theme of differential privacy in metric spaces, Feyisetan et al. (2019a) introduce a noise distribution tailored to the dχ-privacy criterion, which calibrates ϵ values based on the geometric properties of the word embedding space to maintain plausible deniability of original words. This methodology was tested across various tasks including binary sentiment classification on the perturbed IMDB dataset, multi-class classification on Enron, and question answering on InsuranceQA, revealing less than 2% utility loss in binary classification tasks, even at low ϵ values, indicating a favourable balance of privacy and utility compared to other methods like Versatile (Arampatzis et al., 2015) and Incognito (Masood et al., 2018).

The Differential Privacy mechanism, in textual examples, should not disrupt the syntactic structure of a sentence. For this reason, Krishna et al. (2021) developed ADePT, an auto-encoder-based differential privacy algorithm aimed at anonymizing text while preserving its utility. This method involves clipping and noise introduction—either Laplacian or Gaussian—between encoding and decoding to form a differentially private latent representation. They hypothesize that data regeneration via an auto-regressive decoder maintains the syntactic structure of the sentence. Since they focus on the anonymization for Intention Classification (IC), an intention label is included before encoding to facilitate training an annotation-aware auto-encoder. Despite ADePT's design to enhance privacy against Membership Inference Attacks and its superiority over traditional privacy-sensitive word replacement methods like Feyisetan et al. (2019b), Habernal (2021) point out that ADePT actually fails to achieve true differential privacy and exhibits higher sensitivity than reported, casting doubt on its effectiveness. In a similar fashion, Weggenmann et al. (2022) introduce a DP-VAE (Differentially Private Variational AutoEncoder) that also perturbs the latent space to secure privacy, following the model depicted in 5. This model contracts the mean vector of each latent representation towards the latent space's origin before adding isotropic Gaussian noise. Applied to anonymize IMDb and Yelp reviews, this technique successfully reduces authorship attribution accuracy significantly while only slightly impacting sentiment classification accuracy. Further testing with a GPT2-based model confirms that the anonymized texts remain coherent and understandable, despite a slight increase in perplexity.

Also transformer-based models can be used in this setting. Extending the research on Differential Privacy for text privatization in this direction, Igamberdiev & Habernal (2023) introduced DP-BART, a privatization model using the BART architecture instead of an LSTM, employing a similar strategy of clipping and noise addition in the latent encoder output, as seen in 5. DP-BART tackles the challenges of high sensitivity and computational demands associated with transformers through a novel clipping technique (by value) and an iterative pruning and training method that simplifies the encoder's output vectors. Tested against ADePT on datasets like ATIS, Snips, IMDb, Drugs.com, and Amazon, DP-BART demonstrated superior performance in both privacy and utility. Another transformer-based approach is proposed by Du et al. (2023a), where

they use BERT and propose the Purkayastha Mechanism to take advantage of the angular distance as a metric. They also propose to sanitize labels when too sensitive, using Randomized Response if the label space is small enough. Otherwise, they prune the label space leveraging prior knowledge and then use RR. This approach is tested on SST-2, IMDb, and QNLI datasets, representing tasks for sentiment analysis and question-answering based on movie reviews and Wikipedia content, respectively. Performances are close to the non-private version with an increased robustness to MIAs.

Levaraging the Exponential Mechanism Yue et al. (2021) argue that – due to the curse of dimensionality – directly injecting noise in large high-dimensional word embedding may fail to strike a nice privacy-utility balance, since it becomes exponentially less likely to find a noisy embedding close to a real one on every dimension. Yue et al. (2021) developed two strategies, SANTEXT and SANTEXT+, to replace original text tokens with "sanitized" tokens via the exponential mechanism. Both methods utilize embeddings to map words to compute their semantic distance. Then, once a word is identified to be replaced, the target word is selected with a probability that is inversely proportional to the distance with respect to the original one. SANTEXT indiscriminately replaces any word, often resulting in excessive utility loss and protection, while SANTEXT+ specifically targets and replaces the top w low-frequency tokens, identified as most likely to be private. These approaches enforce Metric Local Differential Privacy (MLDP), maintaining utility by favouring tokens closer to the original, though requiring a high epsilon value, like 16, to achieve notable improvements in defence rate. Similarly, Chen et al. (2023a) introduced CUSTEXT, another MLDP approach where every token is replaced using a mapping function that determines the output set for an input token and a sampling function based on the exponential mechanism for selecting replacements. The main difference here is that every input embedding has a custom set of output embeddings, which is a subset of the full output space used in SANTEXT, and thus, it reduces the chances that the output is semantically irrelevant to the input. They claim the best privacy/accuracy tradeoff for these systems so far, achieving good results with relatively low epsilon values (up to 3), although there is still a notable performance drop compared to using plain text.

SANTEXT and CUSTEXT, while being effective, show some limitations. Tong et al. (2024) critique the small adjacency list used in SANTEXT and CUSTEXT, which could expose the methods to embedding inversion attacks and inefficacy in open-ended text generation, where bias matters more. Tong proposes RANTEXT, which dynamically generates a "random adjacency list" for each token then adds noise to the token's embedding and selects nearby tokens based on the noisy embedding, leveraging the Laplace mechanism to ensure Differential Privacy. This method, developed within a framework for using black-box LLMs in private inference, includes a denoising module to reconstruct private outputs. RANTEXT, tested on CNN/Daily Mail (Hermann et al., 2015), Wikitext-103-v1 (Merity et al., 2016) and the ArXiv dataset (Clement et al., 2019), generally outperforms other methods in both utility and privacy, with exceptions in some areas where CUSTEXT+ performs slightly better. This comprehensive approach enhances the robustness of textual differential privacy, especially against inference and embedding inversion attacks.

Beyond the exponential mechanism Carvalho et al. (2021) critique the exponential mechanism's fit for Metric Differential Privacy, noting it fails to consider the density of the space from which outcomes are selected. They propose the Truncated Exponential Mechanism (TEM), which adjusts noise addition based on the density of the embedding space around an input word, evaluated for utility with a sentiment classification model and privacy with MIAs, showing substantial improvements over the baseline (Feyisetan et al., 2019b). Adding to these developments, Xu et al. (2020) propose a Regularized Mahalanobis Metric for text perturbation to address the shortcomings of traditional noise injection methods that suffer from low utility due to the uniform application of noise in continuous embedding spaces. By using a regularized version of the Mahalanobis metric to add elliptical noise, accounting for the covariance structure in the embedding space, this method aims to ensure sufficient likelihood of replacement for words in sparse regions without sacrificing overall utility. Tested on Twitter and SMSSpam datasets, this method shows a lower number of unchanged words and a greater diversity of word substitutions compared to the Laplace mechanism, suggesting higher privacy while maintaining comparable text classification performances. These advancements highlight the ongoing challenges and innovations in achieving effective textual differential privacy without compromising the utility of the data.

Known Issues Expanding on the theme of embedding-based perturbation methods, Xu et al. (2021) acknowledge a common limitation in such approaches, where even minor noise might still leave the original embedding as the nearest neighbour. To counter this, they introduce a method inspired by Vickrey auctions to balance the selection between the first and second closest neighbours using a tuning parameter. This parameter is optimized through a constrained optimization problem aiming to maximize utility while ensuring privacy guarantees. Conducted experiments on Product Reviews, IMDb Movie Reviews, and Twitter datasets reveal that this Vickrey mechanism can enhance utility by up to 50% on real text classification datasets compared to existing methods, without compromising privacy, highlighting an innovative stride in balancing the privacy-utility tradeoff in textual differential privacy strategies. Further examining the methods in the field, Arnold et al. (2023) critique the focus on semantic-based embeddings in existing Metric Differential Privacy techniques, noting their failure to consider syntax specifically. Analyzing these techniques' ability to preserve grammatical properties, they found that privatization mechanisms often privilege nouns even when replacing other grammatical categories. To address this, they incorporated grammatical categories into the privatization process, using methodologies from Feyisetan et al. (2019a) or Carvalho et al. (2021). Results indicate that this approach can enhance performance on downstream tasks, although it also shows a slight increase in self-substitution, suggesting a small decrease in privacy compared to previous approaches. This development underscores the complex interplay between maintaining linguistic integrity and ensuring robust privacy protection in textual data. Adding to the critique, Mattern et al. (2022b) identify significant limitations of Metric Differential Privacy methods, such as the increased noise necessary for long texts or their inability to rephrase effectively. Proposing an alternative, they suggest a paraphrasing model obtained by fine-tuning GPT2, inspired by Witteveen & Andrews (2019). This model introduces noise by sampling temperature based on the exponential mechanism, aiming to balance privacy with natural language generation capabilities. Tested for its effectiveness using the Matthews Correlation Coefficient (MCC) for authorship attribution and sentiment analysis, as well as semantic similarity and perplexity metrics to assess how well the original meaning is preserved and the quality of the generated text, this paraphrasing model proves superior to traditional word-level DP mechanisms.

5 Solutions for Preserving Privacy in LLMs: Model

In this section, we explore the various contemporary methods employed to ensure privacy preservation within machine learning models. The focus is on four primary approaches that operate directly on the model. Firstly, we explore the concept of Differential Privacy applied during the training phase (§5.1). This method involves training a model with techniques designed to protect the privacy of the data used by injecting noise (for an overview on Differential Privacy see §2.2). Secondly, we examine the application of Differential Privacy during the inference phase (§5.2), where privacy-preserving mechanisms are applied when the model is making predictions. Thirdly, we discuss federated learning with Differential Privacy (§5.3), which combines federated learning techniques with Differential Privacy to provide data safety in decentralized environments. Lastly, we discuss machine unlearning (§5.4), a technique that enables an existing model to forget or unlearn specific privacy-sensitive information, and Homomorphic Encryption (§5.5) a cryptographic method that allows computation on encrypted data without decrypting it. Each of these methods offers unique benefits and challenges in the pursuit of robust privacy protection in large language models.

5.1 Training Large Language Models with Differential Privacy

Training LLMs with Differential Privacy

Definition: Training LLMs with Differential Privacy consists of all the approaches that apply Differential Privacy at training time during one of the training phases of Language Models.

Training a Large Language Model (LLM) typically involves three sequential phases: pre-training on a broad collection of unannotated corpora (phase 1), fine-tuning on a specialized corpus using instruction-tuning procedures (phase 2), and final alignment according to human preferences (phase 3). We refer the reader to §2.1 for an in-depth overview of how these phases work. In this section, we propose methodologies that guarantee privacy by using differential privacy during the pre-training phase (phase 1 - §5.1.1) or apply it

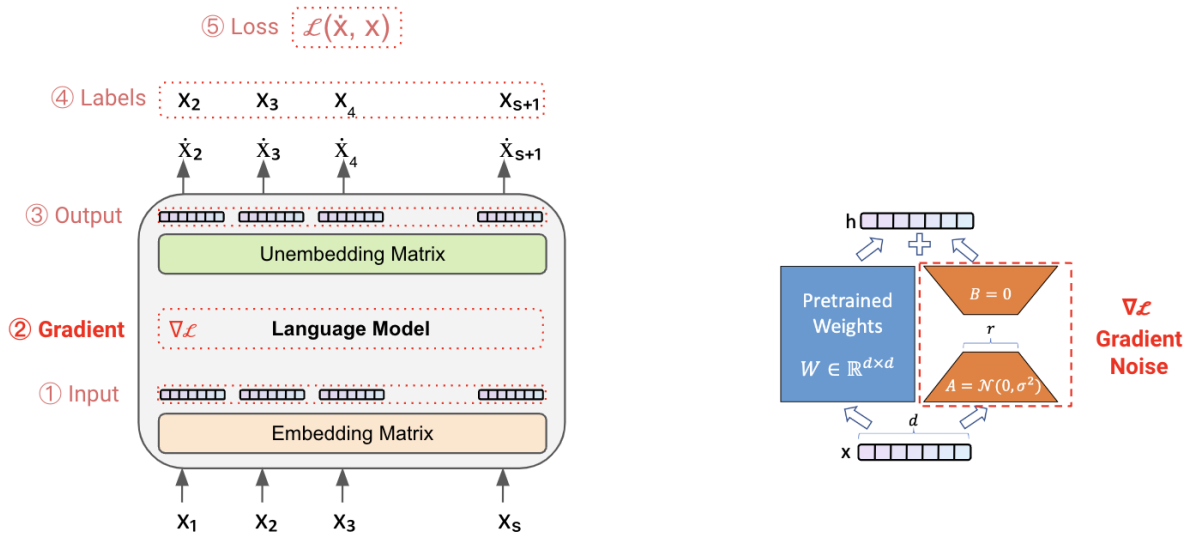

Figure 6: **Differential Privacy Applied on LMs.** Differential privacy represents one of the main solutions used to preserve privacy when training a language model. We have already discussed Differential Privacy (DP) in Section 2.2; here, we contextualize DP with its usage in modern large language models that, most of the time, consist of adding noise in the training gradients (DP-SGD). **On the right:** Adding noise to the gradients of all the training parameters in a language model is often expensive. Parameter-efficient fine-tuning techniques can reduce the number of training parameters, thus lowering the cost of applying Differential Privacy to these gradients. Picture adapted from LoRA (Hu et al., 2021), a popular PEFT method.

during the fine-tuning phase (phase 2 - §5.1.2). We also analyze methods that focus on Parameter Efficient Fine-Tuning techniques (§5.1.3) while leaving out of the scope of this survey works that focus on alignment via Reinforcement Learning and differential privacy since, to the best of our knowledge, there are very few works that just started exploring this aspect (Wu et al., 2024).

Despite the significant concerns over privacy, most of today's LLMs abstain from any privacy-specific measure, relying instead on publicly available internet data for pre-training. While this data is public, it often contains sensitive information, posing risks that might harm individuals and undermine trust. Public data, thus, carries inherent risks, yet remains a widely used resource. On the other hand, fine-tuning often requires using private datasets that entities prefer to keep confidential to protect sensitive information or maintain a competitive edge. We refer to *"public training"* when the training process involves using datasets that are available to the general public and to *"private training"* when the training process relies on restricted, confidential data that is not available to the public and must be carefully protected for privacy reasons. Here, Differential Privacy could offer a solution, minimizing risks associated with data memorization and safeguarding the privacy of individuals in these datasets.

We have already explained Differential Privacy (DP) in Section 2.2; here, we contextualize DP with its usage in modern large language models. Specifically, working with DP often involves deciding where to inject noise into the training process, typically by adding noise to the gradients as shown in Step 2 of Figure 5.1. This approach generally involves using DP-SGD (Song et al., 2013), a variation of the classic SGD that clips gradient norms and then adds Gaussian noise to the average clipped gradient norm (see §2.2 for more details).

In the following sections, we will focus on how Differential Privacy can be applied to modern language models.

5.1.1 Pre-training with Differential Privacy

Pre-training a Large Language Model from scratch is usually a very cost-intensive procedure that just a few companies and institutions in the world can afford. In addition to this, adding differential privacy to this pre-training phase further increases the difficulties since: (i) adding noise increases the already exorbitant training cost, slowing the convergence time, and (ii) the training phase requires a vast collection of private data, which is often scarce and difficult to obtain. Thus, most models pre-trained from scratch with Differential Privacy are relatively small compared to current Large Language Model standards. Nevertheless, they offer both theoretical and practical insights into why fully private pre-training is considered a sub-optimal strategy. Instead, a mixed solution involving pre-training on public data is recommended. In this section, we will explain why. All the works discussed are also summarized in Table 4.

Hoory et al. (2021) conduct a fully private training of BERT, utilizing DP-SGD during both pre-training and fine-tuning stages. The pre-training employs the standard BERT dataset from Devlin et al. (2019), supplemented with the MIMIC-III dataset (Johnson et al., 2016). Fine-tuning is then carried out on the i2b2-2010 dataset, sourced from the i2b2 National Center for Biomedical Computing and the NLP Shared Tasks Challenges, focusing on entity extraction within the clinical domain. The experiments are conducted with and without Differential Privacy, and the outcomes are measured by the F1-Score. Their findings indicate only a minor performance decrement with privacy-preserving methods, thus demonstrating a favourable privacy/utility tradeoff.

Yin & Habernal (2022) introduce a novel approach to fully pre-train BERT from scratch using a specialized legal tokenizer. They first perform conventional pre-training using BookCorpus and Wikipedia, followed by an additional pre-training phase using DP-SGD on a legal corpus, emphasizing their focus on safeguarding the privacy of the pre-training data. Subsequently, the model undergoes non-private fine-tuning on two downstream tasks: binary classification on the Overruling dataset (Zheng et al., 2021) and multiple-choice QA on CaseHOLD (Zheng et al., 2021). These steps highlight the authors' intent to protect only the data used during the DP-SGD pre-training phase. The outcomes of this approach not only match but sometimes surpass those of non-private benchmarks, demonstrating the feasibility of maintaining high utility while employing privacy-preserving methods. However, the study lacks a specific privacy benchmark to assess any potential privacy leakage.

It is crucial to acknowledge that the discussed works were capable of conducting full training from scratch largely because they utilized BERT (Devlin et al., 2019), which possesses a relatively modest 109M parameters, a figure significantly smaller than those of modern LLMs such as GPT-3 with 175B parameters (Brown et al., 2020). Currently, the standard practice, even in settings where data are not private and there is no need to perform DP tuning, involves leveraging a pre-trained model and fine-tuning it for specific tasks. This approach not only reduces computational demands but also allows the model to harness a broad and diverse initial training dataset, enabling it to subsequently adapt more effectively to a private dataset and its specific domain, often enhancing performance. However, the benefits extend beyond performance improvements. Recent studies, including Kerrigan et al. (2020) that compare public pre-training with differentially private fine-tuning against entirely private DP training, demonstrate significantly better results with the former method. This provides empirical evidence that DP learning greatly benefits from an initial phase of public pre-training. Full training solely on a private dataset often results in high perplexity values, indicating a model that would likely be ineffective. In Kerrigan et al. (2020), the comparison of models fully trained privately versus those benefiting from public pre-training followed by private fine-tuning demonstrates that public pre-training is essential for enabling effective private learning. Additionally, since DP-SGD introduces noise during training, limiting this noise to the fine-tuning phase rather than the entire training process results in less overall noise and better performance. This improvement is due to the general (clean) knowledge acquired during the pre-training phase.

Ganesh et al. (2023) present a study that compares the performance obtained via a public pre-training followed by private fine-tuning with a scenario where both stages are conducted in a differentially private manner. This paper emphasizes the theoretical underpinnings of Differential Privacy in training models, arguing that for a differentially private model to be effective on the task it is trained on, it is essential to first undergo pre-training on public data. The theoretical basis provided in the study explains that optimization

involves two phases: initially, the model transitions from a random initialization towards a favourable basin, and subsequently, it addresses a convex optimization problem to find the local minimum within that basin. If attempted entirely within a private context, incorporating private data and noise addition, the initial stage would necessitate substantially more data to achieve comparable outcomes to those achievable with public data, as supported by experimental evidence. Moreover, given the typical scarcity of private data, *it might be impractical or even impossible to gather sufficient data to effectively train a model privately from scratch.* Therefore, pre-training on a larger public dataset not only aids in overcoming these limitations but also enhances the model's resilience to distribution shifts between the public and private datasets.

Tramèr & Boneh (2020) offer another perspective, arriving at a similar conclusion: public pre-training is crucial for achieving desirable outcomes in a private learning environment. Although this study is centered on image classification using CNNs, the findings reinforce that the advantages of public pre-training transcend various contexts, including this one. In their approach, the researchers enhance the performances of fully private training by incorporating handcrafted features, which yield improved results. Nevertheless, these outcomes still do not match those achieved through a combination of public pre-training and private fine-tuning. **The collective insight from numerous studies is that a public pre-training phase is indispensable in private learning scenarios.**

While the works reviewed so far performed a "classical pre-training" (i.e., gaining knowledge from a large dataset), Yu et al. (2023) demonstrate that pre-training can also be conducted selectively, i.e., by focusing on a specific (smaller) subset of data that allows DP training to become more efficient. Their approach involves a novel technique of selecting a subset of the pre-training dataset, OpenWebText (Gokaslan & Cohen, 2019), whose distribution closely resembles the fine-tuning dataset's distribution, Enron email dataset (Klimt & Yang, 2004). To filter the pre-training dataset, a classifier trained with Differential Privacy predicts whether a pre-training sample belongs to the distribution of the fine-tuning (private) dataset. Each sample in the pre-training dataset is then scored for how likely it is to belong to the fine-tuning dataset, then the top-k samples are used for public pre-training. This is followed by differentially private fine-tuning on the private dataset, with the final privacy loss taking into account both the classifier and the ultimate model. This method of dataset selection is designed to minimize distribution shifts, enabling smaller models to achieve better results. The results demonstrate strong performances, surpassing more complex baselines and thus enhancing efficiency to the point where this can be considered a model compression tool. However, a potential oversight of this method is the non-consideration of biases within the fine-tuning dataset, which could lead to the inadvertent transfer of these biases into the model through the selected pre-training samples. This is not as likely when the pre-training dataset is fully utilized.

5.1.2 Fine-tuning with Differential Privacy

Pre-training a large language model from scratch, with or without Differential Privacy, has become wildly expensive given the sheer size of these models. Currently, the prevalent approach consists of using an already available LLM pretrained checkpoint and fine-tuning it for specific downstream tasks, including applications of Differential Privacy (DP). In this section, we survey several different works that apply Differential Privacy on top of already pre-trained LMs. These works are also available in tabular format in Table 5. Yue et al. (2022) show that using DP-SGD it is possible to successfully train LMs for text generation. They apply DP-SGD to fine-tune various sizes of GPT2, using the Open Yelp Dataset. This is mainly done to generate synthetic reviews that cannot be traced to the actual reviewer. The generated reviews are then used to train classifiers to predict ratings and business categories. The quality of the synthetic text is assessed by comparing its distribution to the original dataset (through F1-score, Fréchet Inception Distance Heusel et al. (2018), and MAUVE Pillutla et al. (2021)) and the classifier performance against models trained on the original data. In this task, the largest GPT2 models produce data that more closely resemble the original data, probably because they are less influenced by DP. Experiments indicate that DP introduces a minimal performance drop, often less than 0.01%, within a privacy budget of $\epsilon = 4$. Remarkably, the GPT2-Large model with DP nearly matches the performance of its non-private counterpart, with only a 0.008% difference in accuracy on downstream tasks. Privacy testing was performed following the method by Carlini et al. (2019): this consists of injecting canaries inside the training set in order to extract them later on, to prove that the model is capable (or not) of memorizing training data. When these canaries were injected into

the training data, the model could not be prompted to reproduce them under DP conditions, even with a repetition rate of 100. In contrast, without DP, the canaries were reproduced 80% of the time, demonstrating effective privacy preservation. This work highlights the feasibility of DP fine-tuning. When we investigate DP learning and the used techniques, there is a constant comparison with standard learning, as it is the most widely known and adopted. Papernot et al. (2019) specifically focus on the differences between private and public training, trying to shed some light on the best conditions for private training. They investigate architectures, parameter initializations and optimizers. We are not very interested in architectures as they do not consider transformers but only convolutional architectures and we already know from §5.1.1 that it's best to start from parameters that were previously optimized through a public pre-trained. The novelty that this research points out is that adaptive optimizers like Adam provide marginal gains (if any) in this context. Putting in direct comparison DP-SGD and DP-Adam, they find that while the latter converges faster initially, it eventually slows down to the point where DP-SGD can outperform it, reaching a lower loss. Another relevant difference they point out is how different a working learning rate is when tuning DP-SGD with respect to standard SGD (with all the other conditions being the same). Definitively this work highlights how DP-tuning, while effectively working, requires to be handled in a slightly different way than a standard tuning.

Li et al. (2021), building on the work of Papernot et al. (2019) on how to approach private-tuning, explore the effects of hyperparameters and fine-tuning objectives, identifying typical pitfalls of DP tuning when it's treated as normal tuning. Their findings are particularly relevant as they suggest that, with proper execution, the performance of DP-tuned models can rival strong non-private benchmarks. They introduce a technique known as *Ghost Clipping*. Ghost Clipping is an important step in the landmark of private tuning because it greatly reduces memory usage, aligning the efficiency of private learning with that of non-private learning. Before that, DP-tuning was much slower and heavier than normal tuning to the point where it was almost not feasible (especially for large batch sizes, which are necessary for models like BERT) and even if it was, it would reach much lower performances. Ghost Clipping is an expansion of the concept described in (Lee & Kifer, 2020), it avoids the need to instantiate per-example gradients for each linear layer by handling one layer at a time and necessitating an additional backpropagation pass per batch. Extensive testing across various models—RoBERTa for natural language understanding (NLU), GPT2 for natural language generation (NLG), and both GPT-2 and DialoGPT for chit-chat dialogue generation—reveals that private models perform nearly as well as public ones, with only minor differences in perplexity and BLEU scores, showcasing competitive results. When compared to using vanilla Opacus, Ghost Clipping shows a 3x improvement in efficiency. While Ghost Clipping is already a big step in efficiency, Ding et al. (2024) point out that this method cannot be applied to standard Transformers because of non-plain feed-forward (and back-propagation) topology caused by embedding sharing that hinders Differential Privacy guarantees. The authors proposed *Phantom Clipping*, which is able to handle parameter sharing thanks to the manipulation of tensor operations. This ensures that the shared parameters' gradients are computed and clipped without redundancy, hence preserving the Differential Privacy guarantees. They also show and solve another problem typical of DP-tuning: attention distraction. Attention distraction is a phenomenon that occurs when noise added in DP-SGD to maintain privacy distorts the self-attention mechanism, causing the model to inaccurately weigh parts of the input data. This leads to less effective learning as the model might focus on irrelevant details or overlook important information. To address this, the proposed "Re-Attention Mechanism" recalibrates attention scores by tracking and adjusting for the variance introduced by the noise. This adjustment ensures that attention scores more accurately reflect the true underlying data relationships despite the noise, thus maintaining the model's performance and efficiency under differential privacy constraints. *Ghost Clipping* and *Phantom Clipping* are already a big step towards making private learning as efficient as standard learning, but there is still a gap to close. In order to get even closer, Bu et al. (2022b) advance the methodology proposed by Li et al. (2021) with a novel Book-Keeping (BK) technique. This new method requires only a single back-propagation round and does not necessitate the instantiation of per-sample gradients. As a result, it significantly enhances throughput compared to Ghost Clipping while maintaining a similar memory footprint. This means that this new technique is a big step forward as it offers a more scalable solution for handling data of higher dimensions.

It is often possible that there is some overlap between pre-training data and fine-tuning data. While this is usually not an issue, in a private setting, where only fine-tuning is done privately and pre-training is

standard, this kind of contamination would result in a data leakage that could expose sensitive data. In this sense, the work done by Kurakin et al. (2024) is important as it addresses the potential data contamination between pre-training and fine-tuning datasets. To tackle this issue, they employ an LLM based on LaMDA (Thoppilan et al., 2022), initially applying deduplication techniques (Lee et al., 2022) to ensure fine-tuning data are excluded from the pre-training dataset. Following deduplication, they proceed with standard pre-training and subsequently conduct private fine-tuning using both full and Parameter Efficient Fine-Tuning (PEFT) methods. The model is then utilized to generate a synthetic dataset for a downstream sentiment analysis task, employing BERT and WordCNN for classification. Although the results are good, showing only minor performance drops for DP-tuning with respect to standard tuning, it is important to note the use of relatively high privacy budgets (e.g., ϵ=10). Moreover, this approach is tested only on binary classification tasks, as opposed to the more complex multi-class classification challenges addressed by Li et al. (2021).

Mattern et al. (2022a) introduce a variant of DP fine-tuning by applying it to GPT2 for generating private versions of IMDb and Amazon reviews datasets, which are then used to fine-tune models for classification. Following the methods of Schick & Schütze (2020), during the fine-tuning of GPT-2 they enhance sample generation fidelity by incorporating instructions based on the original dataset's characteristics, such as sentiment or category, into templates. Since the datasets are made for classification, the data is labelled. The idea is to use these labels to steer the generation, also during training, by asking specifically for a sample corresponding to a label, like, for example, asking for a "positive" review. This approach integrates classification labels directly into the LLM's fine-tuning prompts to prevent generating mismatched samples (like a positive review that reads like a negative one) and introduces a novel loss function that penalizes such mismatches. After training, the model samples a subset of the original training set, which is then utilized for further fine-tuning on binary classification tasks. For comparative analysis, the authors also fine-tune models using DP on the actual data. The main result is the demonstration that models trained on synthetic data generated via DP and then fine-tuned in a non-private manner outperform those trained directly on real data with DP. Additionally, the researchers assess data privacy by examining duplicates between original and synthetic datasets, noting a significant reduction, with duplicates completely eliminated in some instances. In this study, similarly to what was shown by Kurakin et al. (2024), the downstream task is binary classification, which is not particularly hard.

Differential Privacy (DP) introduces various challenges, notably due to the addition of noise. One such issue, the increased memory consumption from noise addition, is addressed by Ghost clipping (Li et al., 2021). To further mitigate these challenges, Behnia et al. (2022) introduce a new framework designed to reduce the overall amount of noise required to maintain privacy guarantees. This framework utilizes the Edgeworth accountant (Wang et al., 2022), which provides a "finite sample guarantee" unlike the typically asymptotic nature of DP guarantees. This is particularly advantageous for fine-tuning an LLM where the number of tuning steps is relatively limited. By calculating the necessary noise levels based on privacy parameters and the number of compositions for a specific dataset, this new framework significantly reduces noise impact—by up to 5.6%. Additionally, it achieves a performance improvement of up to 1.1% across various natural language understanding (NLU) tasks compared to previous benchmarks. This makes the approach relevant as it not only enhances privacy but also improves efficiency in DP applications.

Although pretraining on public data followed by fine-tuning on private data has long been considered the standard and safest method for private model training, recent evidence suggests that even this approach carries certain risks. Feng & Tramèr (2024) recently showed how privacy backdoors can be injected in pre-trained models making them susceptible to different kinds of attacks. This research explores how neurons can be subtly manipulated to retain their primary function while gaining malicious capabilities. By enabling neurons to hold a gradient for a single input and "deactivate" to prevent overwriting during subsequent training, the model becomes vulnerable to gradient inversion attacks, similar to those used in federated learning models (see 3.3 for details). While such attacks typically require white-box access, the researchers also demonstrate attacks on black-box models. Specifically, they enable membership inference attacks (MIAs, see 3.2) by modifying units in the model's second-to-last layer, which are uniquely activated by specific training data points. By adjusting the model's activation, they can determine whether a sample was part of the training set by comparing logits before and after fine-tuning. Although the study only tests on MLPs, ViT, and BERT, and it remains unclear if these vulnerabilities apply to LLMs, it highlights a significant

risk in the public pre-training and private fine-tuning paradigm: if the pre-trained model or its provider is untrusted, loose privacy budgets commonly assumed in the literature are unsafe. Any efforts to establish tighter privacy guarantees must account for the model's integrity.

5.1.3 Parameter-Efficient Fine-Tuning with Differential Privacy

Modern large language models (LLMs) often contain billions of parameters, making even full fine-tuning impractical due to high computational demands, particularly when paired with noise injection for Differential Privacy (DP). Instead, Parameter Efficient Fine-Tuning (PEFT) methods like adapter tuning (Houlsby et al., 2019), compacter tuning (Mahabadi et al., 2021), prompt tuning (Lester et al., 2021), prefix tuning (Li & Liang, 2021), and notably LoRA (Hu et al., 2021), have become more prevalent. Around the same time LoRA came out, also Reparametrized Gradient Perturbation (RGP) was introduced (Yu et al., 2021b), although with less success. RGP is a PEFT technique specific to the field of DP. It enhances efficiency by reconfiguring each weight matrix into smaller matrices, optimizing gradient storage and computation. These techniques, which involve adapting a subset of the model's parameters or incorporating external parameters, facilitate faster training and enhanced modularity. They allow the model to manage various tasks without requiring complete re-training. Remarkably, even with fewer parameters being updated, these methods often achieve performance that is comparable to or even surpasses full fine-tuning, depending on the specific task and dataset (Sun et al., 2023; Suri et al., 2023). In this section, we survey methods that apply Differential Privacy with PEFT techniques in order to reduce the computational burden of fine-tuning while keeping privacy standards. A synthetic overview of these methods is available at 5.

Yu et al. (2021a) applied all the PEFT techniques just mentioned with DP to RoBERTa and GPT2 without any particular challenge. As a matter of fact, they achieve performances nearly equivalent to their public counterparts, especially with RoBERTa where many PEFT techniques with DP notably outperformed full fine-tuning. The study also highlights the efficiency of DP when combined with LoRA in terms of speed and memory use, underscoring the higher viability of private learning in a PEFT framework. This is done by using DP-SGD with LoRA

In a continuation of similar research, Li et al. (2021) conduct tests with both RoBERTa and GPT2, employing DP-Adam, which is similar to DP-SGD but based on Adam, for updates and moment accumulation) and the Ghost Clipping technique. This introduction reduces memory usage, making the difference in efficiency between private and non-private learning negligible (see §5 for more info). At the same time, also the performance drop is proven to be negligible, making DP-PEFT a viable option. GPT2 is evaluated in tasks like table-to-text and chit-chat generation, where the performance of full fine-tuning and PEFT proves to be closely matched, with full fine-tuning showing a slight edge both with and without Differential Privacy. This indicates that PEFT can match the efficacy of full fine-tuning even in a private setting. Additionally, the results challenge previous assumptions seen in the work of Yu et al. (2021a), where PEFT's superior performance was attributed to lesser noise addition. The findings here suggest that higher noise levels in full fine-tuning might not detrimentally affect the privacy/utility tradeoff as previously thought. This observation prompts further inquiry into the mechanisms of private learning, offering a pathway to deeper insights into how different fine-tuning methods impact model performance under privacy constraints.

While LoRA remains the predominant technique for private tuning, alternative methods also demonstrate effectiveness. Li et al. (2023d) introduce RAPT, a privatization framework designed for prompt-tuning and prefix-tuning on a model that does not necessarily need to be locally run. This is made specifically to make it possible to privately train a remote model through prompt tuning APIs, like Nvidia NeMo. This involves adding noise to the prompts before they are sent, addressing the issue that direct prompt tuning on privatized data can significantly impair downstream performances. To counteract this, they propose a novel privatized token reconstruction task, which is trained concurrently with the downstream task. The experiments utilize BERT and T5 models across various datasets such as SST (Socher et al., 2013), QQP (Lili Jiang, 2017), and TP-UK (Hovy et al., 2015), with adjustable privacy parameters. Privacy is assessed through the defensive capabilities against both an embedding inversion attack (see §3.3.1) and an attribute inference attack, where both prompt-tuning and prefix-tuning show enhanced robustness. Utility tests reveal that performances suffer without the use of the privatized token reconstruction, which significantly boosts performance and confirms its efficacy in helping LLMs develop better representations.

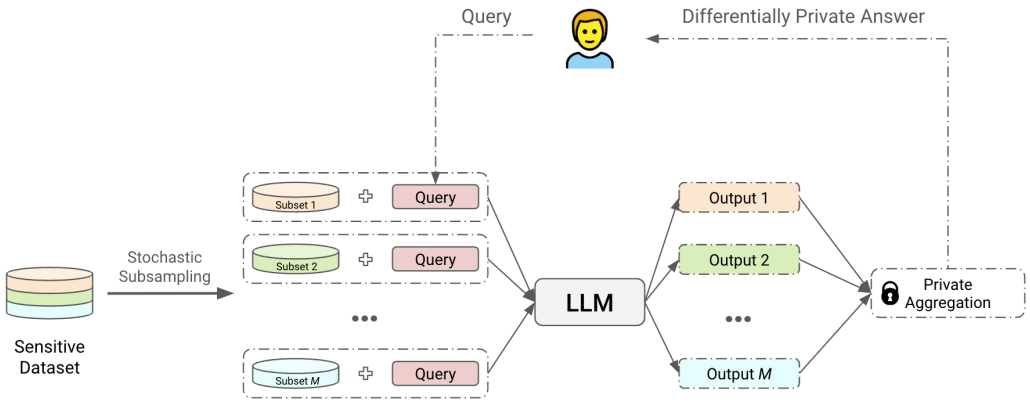

Figure 7: **Differentially Private In-Context Learning.** High-level description of applying Differential Privacy during in-context learning: the original dataset is used to create several subsamples via a stochastic sampling procedure; these subsamples are then combined with the user query (prompt) and fed to an LLM; all the generated outputs are aggregated in a private fashion, producing a single differentially private prediction.

5.2 Inference with Differential Privacy

Inference with Differential Privacy

Definition: Inference with Differential Privacy consists of all the approaches that apply Differential Privacy at inference time without changing internal parameters. DP is used in two classes of DP-inference approaches: differentially private in-context learning (ICL) or noising-denoising external modules.

The first class of DP-inference models is based on In-context learning (ICL) (Brown et al., 2020) (see §2.1). Indeed, ICL may be used to increase privacy in LLMs since private data can be given to LLMs only at inference time, and thus, this private data should not be included in generalistic LLMs and can stay under the control of the owners holding the data rights. For example, if a medical decision on a specific patient should be taken by considering all patients in a specific hospital, these patients can be given as part of the prompt in in-context learning (ICL) approach. However, nothing guarantees that a malicious user can try to extract this personal data by using specific prompts following the general ICL prompt (see §3.1.2). This is an active area of research that aims to embed Differential Privacy mechanisms in inference, performing what is called *differentially private in-context learning*. The work presented here all follow a similar high-level structure represented in Figure 5.2: the original dataset is used subsampled or partitioned in several subsets; the subsamples are then combined with the user prompt and fed to an LLM; all the generated outputs are aggregated in a private fashion, producing a single differentially private prediction.

To address the issue rising with ICL, Wu et al. (2023a) introduce a novel approach, called Private in-context learning, to ensure privacy in ICL operations by partitioning the private dataset into M subsets, each containing N samples. These subsets are used as N-shot examples for generating M different ICL prompts, which in turn produce responses for the next token prediction or output classification. Results from each prompt are then privately aggregated using techniques tailored to the specific task: (1) Report-Noisy-Max with Gaussian noise for text classification that ensures differential privacy; (2) aggregation methods for text generation. Two aggregation methods are explored: Embedding Space Aggregation, where outputs are mapped to an embedding space to compute and noise the mean output, followed by querying the LLM multiple times to find the closest match with cosine similarity, and Keyword Space Aggregation, which involves noisy top-k counting of words across outputs using mechanisms like the Joint Exponential Mechanism (Gillenwater et al., 2022) or Propose-Test-Release (Zhu & Wang, 2022). Despite operating within a low

privacy budget (e.g., $\epsilon = 1$), Private in-context learning (ICL) achieves performance comparable to non-private methods across various tasks such as text classification, document question answering, and text summarization. This method not only demonstrates minimal performance drop with added privacy but also offers a less computationally intense alternative to fine-tuning with DP, providing a practical, lightweight solution for enforcing privacy in LLMs.

Tang et al. (2024) added another security level on top of the method of Wu et al. (2023a): their method generates samples with DP before giving the input as in-context learning to LLMs. Then, the rest is the same, that is, possibly dividing the private dataset into M subsets containing the N samples. Clearly, Tang et al. (2024) show that the distortion of the input private data samples has little influence with respect to target applications. The effectiveness of this method is examined over benchmarks on standard Text Classification and Information Extraction datasets. Performance results are presented as the mean and standard deviation across five runs. The private models demonstrate performance on par with non-private models, indicating that it is feasible to achieve private learning without sacrificing accuracy. The study also evaluates the performance of Gaussian and Exponential mechanisms in the aggregation phase, with the Gaussian mechanism consistently showing slightly better results. This reinforces the viability of using synthetic data for privacy-preserving ICL applications.

To improve the level of security in these ICL-based approaches, Duan et al. (2024a) propose to use a teacher-student approach where real data are given to the teacher model in a way similar to Wu et al. (2023a). The teacher model is then used to give synthetic examples for the ICL of the student model. In this way, the student model, which is exposed to final users, does not have access to real data. The method is called PromptPATE and follows the same general workflow of the standard Private Aggregation of Teacher Ensembles (PATE), presented by Papernot et al. (2018): first, an ensemble of teacher models is trained, then private knowledge transfer is performed, and lastly, a student model is trained. All this training has to be intended as ICL. In this case, instead of training the teacher models, the private dataset is used to generate a set of disjoint prompts for the LLM, so teacher models (or the "flock of stochastic parrots") are actually models performing ICL with disjoint parts of the original dataset. The private knowledge transfer is basically a majority voting from the teacher models made in order to choose the next token prediction. The noisy majority voting is done with the Confident GNMAX algorithm (Papernot et al., 2018) just like in the standard PATE. The chosen token is added to the public input to create a single, differentially private example that can replace private data for ICL on a student model so that the student model does not actually need to undergo real training but can take advantage of the elasticity of prompting. Similarly to Tang et al. (2024), this approach also involves generating synthetic samples for later use in ICL. This method, too, achieves performance comparable to non-private methods, even under stringent privacy conditions.

Hong et al. (2023) present a similar work, with a different aim, which is prompt engineering, and some differences in the aggregation mechanism, called DP-OPT (Differentially Private Offsite Prompt Tuning). In this approach, an ensemble of local models receives disjoint subsets of data to generate the next token. The Exponential mechanism is then employed to privately select the token that appears most frequently. Following this, a Deep Language Network, as described by Sordoni et al. (2023), is used to identify the optimal candidate prompt, applying the Exponential Mechanism again to ensure confidentiality. A significant advantage of this method is that the LLM-engineered prompts are transferable, often yielding results on par with non-private alternatives, depending on the task and the underlying model. The primary distinction here is that DP-ICL is utilized to develop prompts based on training data without any data leakage. This technique demonstrates competitive results on downstream tasks and effectively transfers across several models, including GPT3.5, Llama2, and Vicuna.

The second class of DP-inference models is introduced by Mai et al. (2023). Their method involves partitioning the network into three components: a local encoder responsible for adding Laplacian noise to the data, a remote (often black-box) model such as GPT2, T5, BERT, or Llama2 that processes the privatized embeddings and returns the output, and a local denoising module that refines the output. This configuration enables the use of black-box models while maintaining privacy by adding local preprocessing and postprocessing steps. Since the text is perturbed locally, Differential Privacy is applied on the input, which corresponds to 1) in Figure 5.1 (DP on input). Tested on various downstream tasks, this method shows promising performances, closely trailing behind its non-private counterparts. Additionally, the study conducted inversion

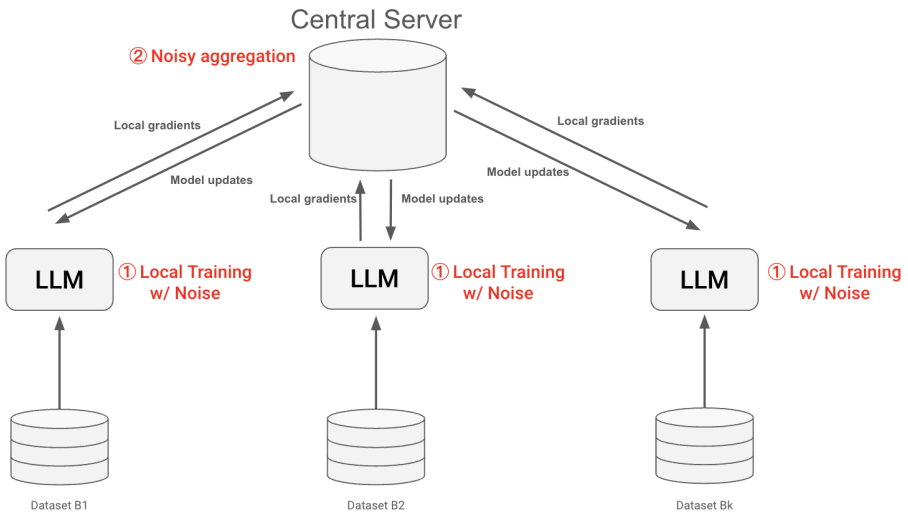

Figure 8: **Federated learning with Differential Privacy.**Differential Privacy in a federated setting can be applied according to 2 different models: 1) in the Local model, local gradients are clipped and noised before being sent to the central server; **or** 2) in the Central model plain gradients are sent to the central server which aggregates them in a noisy manner.

attacks to assess robustness, revealing that BERT is particularly resilient at lower noise levels, GPT2-XLarge performs better under higher noise conditions, and T5 appears to be the most susceptible, requiring stringent privacy settings to ensure safety.

Similarly, Tong et al. (2024) propose InferDPT, which includes a local perturbation algorithm called RANTEXT (discussed in §4.1.2). The perturbed text is sent to a remote black-box LLM, and a local extraction module — a smaller model — processes the perturbed output, reconstructing it to ensure coherence, consistency, and alignment with the original prompt. Users then combine the original prompt and the perturbed generation results to produce the final output. InferDPT, using the RANTEXT mechanism, achieves text generation quality comparable to non-private GPT-4, maintaining high utility despite the added privacy-preserving perturbations.

5.3 Federated Learning with Differential Privacy

Differential Privacy and federated learning are two complementary techniques in privacy-preserving machine learning that tackle different aspects of data security. Federated learning (§2.3) is a distributed computation method that enables learning on private data from various sites without transferring the data offsite. However, this alone is not sufficient for data security, as gradient inversion attacks can potentially recover the original training data by intercepting gradients sent from clients to the server (check discussion in §3.3.2). While federated learning helps mitigate memorization issues (Thakkar et al., 2020), real privacy guarantees require Differential Privacy. In federated learning, Differential Privacy can be implemented in different ways: Figure 5.3 illustrates its application either locally (local Differential Privacy) or globally (central Differential Privacy).

A clarification is needed: central and local models of Differential Privacy (DP) are not exclusively related to federated learning (FL). In central Differential Privacy, perturbation is applied at a global level, typically during aggregation. In contrast, local Differential Privacy applies perturbation locally before the data leaves the site. While DP learning often aligns with FL and involves perturbing gradients, DP was initially developed for the statistical analysis of databases, and both central and local models exist independently of learning processes. §C.1 provides several examples of techniques that fall under the local model of DP, where

text perturbation is applied to create a noisy private dataset that can be shared later, while §C contains examples of techniques of central Differential Privacy, where the noise is applied at aggregation time.

Unfortunately, at the moment of writing, the intersection of differential privacy and federated learning did not produce much that is very relevant for LLMs, mainly due to the size of the models, which makes difficult parameter sharing and updating, and due to data dimensionality, which is an important constraint for text. There is still some interesting research we came across, even though not very relevant for LLMs, hence we leave the discussion for that in the appendix at §B.

The most notable example of central differential privacy is Google's implementation in Gboard. Xu et al. (2023b) discuss and analyze twenty Gboard language models trained for Next Word Prediction using DP-FTRL (refer to §B for a thorough explanation), indicating that all future Gboard models will incorporate differential privacy guarantees. In their DP-FTRL setup, model updates undergo clipping, and noise is applied during global model updates, ensuring Central Differential Privacy. Adaptive clipping, which adjusts the clip norm each round by privately estimating the model's norm at a targeted quantile as described by Andrew et al. (2022), was also utilized. However, adaptive clipping did not show a significant improvement and was less robust during experiments with large report goals, sometimes causing catastrophic failure or a lack of progress in the first 1000 rounds. Despite these issues, adaptive clipping can help reduce hyper-parameter tuning when the privacy budget allows. Public pre-training on C4, as explained by Wang et al. (2023), reduced the rounds needed to achieve target utility by about 1000 under the same noise multiplier, thereby enhancing privacy guarantees. Additionally, DP-FTRL was combined with Secure Aggregation, a cryptographic protocol ensuring the central server only accesses aggregated updates and not individual client updates, as described by Bonawitz et al. (2017). While Secure Aggregation supports DP, it introduces considerable computational slowdown and requires different system configurations for large report goals. The techniques for training language models with differential privacy in Gboard have resulted in models with stronger privacy guarantees than some established standards, such as those used by the US Census Bureau. In Europe, these differentially private neural network models outperform older N-gram models in terms of utility. In the US for English and in Brazil for Portuguese, the new models are comparable to or slightly better than their non-private versions.

For the local model of differential privacy, most works (as discussed in §B) use it for telemetry data collection, which means that they use very low dimensionality data. Data and model dimensionality are very problematic in this setting. Luckily, some recent developments can mitigate these problems. There is a very recent work (Liu et al., 2024b) that shows how this model can be applied to an LLM thanks to LoRA. In this work, the authors developed a federated version of LoRA with Differential Privacy, which allows the sharing of a much smaller number of parameters while keeping the privacy guarantee. The method makes use of LoRA to locally train the model, performs clipping and noising at a local level and then uploads the weight updates to the central server. They use this method to fine-tune Llama-7B and ChatGLM-6B for general and medical question answering, respectively, using SlimPajama and a medical dataset crawled from two online healthcare services platforms. The results for several ε and δ show that, once again, there is a degradation in performance when using DP, but it is mostly negligible, especially considering the privacy gain. Unfortunately, in this paper, there is no experiment to concretely test to which extent this method prevents training data leakage.

5.4 Machine Unlearning

Machine Unlearning

Machine unlearning refers to the process by which a machine learning model is able to forget or remove knowledge about certain data points from its training (Cao & Yang, 2015).

Recent regulations like GDPR guarantees the "right to be forgotten", which allows individuals to withdraw their consent and request the deletion of their personal data (Zhang et al., 2024b). Unfortunately, differential privacy (DP) cannot guarantee this because, for a data sample to be truly "forgotten", it must have zero

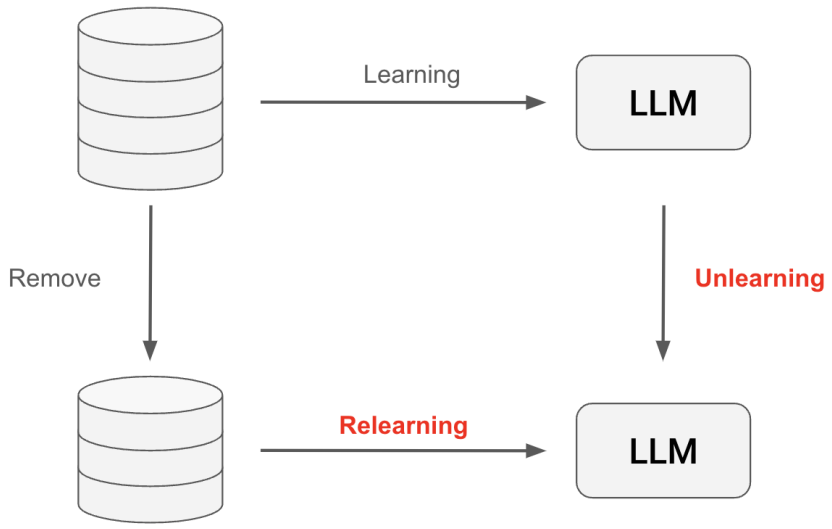

Figure 9: **Machine Unlearning.** Machine Unlearning refers to the process of removing the influence of specific training samples from a machine learning model. This process aims to make the model as if it was never trained on the removed data. Essentially, it ensures that any impact the specified data had on the model's parameters is reversed, effectively erasing its contribution. The image shows that the only real alternative to Unlearning is Relearning, which theoretically works but is extremely impractical for big models like LLMs.

influence on the training process. This requirement translates to a privacy budget of $\epsilon = 0$, which would prevent the model from learning anything.

Machine Unlearning (Cao & Yang, 2015) is an orthogonal field to Differential Privacy, focusing on how to eliminate the influence of specific data samples. The key works related to this topic are summarized in Table 7 in the Appendix. The aim is to make the model equivalent to one trained on a dataset that excluded unwanted entry, as illustrated in Figure 5.4. While test-time prediction to measure the influence of training data have been proposed defining influence functions (Koh & Liang, 2017), this technique does not reveal the impact of individual data point model parameters. Although training strategies designed to ensure unlearning have been proposed and tested for different architectures (Bourtoule et al., 2019), most of the unlearning solutions for LLMs adopt the idea of modifying model parameters after the training phase.

Jang et al. (2022) propose a method focused on unlearning specific information, particularly for language models. This method negates the training loss function – performing gradient ascent – to maximize loss on target sequences, enabling unlearning with minimal performance loss on unrelated sequences. Sequential unlearning proves more effective than batch unlearning. To minimize the impact on performace, Wang et al. (2024b) propose to perform the unlearning only of problematic spans in the sentence, rather than on entire instances. Despite its effectiveness, Eldan & Russinovich (2023) show limitations of gradient ascent in certain scenarios, such as making Llama2-7B forget about Harry Potter. The challenge lies in the high likelihood of several candidate tokens related to Harry Potter, necessitating many gradient descent steps. To address this, they map Harry Potter terms to generic equivalents and then compare predictions from a baseline model with mapped input and a reinforced model trained on Harry Potter with the original input. By reducing the likelihood of tokens predicted by the reinforced model, they fine-tune the baseline model using generic labels. This results in the model losing familiarity with the target knowledge, with minimal leakages resembling secondary knowledge. Performance drops vary by task but remain generally tolerable. Recent findings suggest that gradient ascent should be combined with other techniques to be effective while maintaining model utility: Yao et al. (2024) describe a unified unlearning framework for LLMs, and find that gradient ascent is more robust when combined with gradient descent on a retain dataset.

Chen & Yang (2023) propose creating unlearning layers and training them with a selective student-teacher objective based on KL-divergence. The student model (the additional layers) maximizes divergence from the teacher model (the LLM) on the data to be forgotten while minimizing it on the rest. Different unlearning layers handle different data, but they also provide a method to fuse these layers into one. The model is evaluated on three sets: the Retained Set (where high performance is desired), the Forgot Set (where low performance is desired), and a Test Set. This method yields excellent results, outperforming other unlearning techniques on the Test Set, maintaining raw performance while effectively unlearning the target data.

Unlearning can also be achieved with Reinforcement Learning through an approach called DeMemorization (Kassem et al., 2023). This method takes samples from the pre-training dataset, divides them into prefix and suffix pairs, inputs the prefix, and uses a negative similarity metric to evaluate how different the generated suffix is from the real one. This metric serves as a reward signal, encouraging the language model to develop a paraphrasing policy that generates different tokens to minimize memorization. DeMemorization achieves better performance on downstream tasks because it doesn't remove samples from the dataset. However, for the same reason, it shows less robust privacy performance compared to actual unlearning methods, though the privacy performance remains high.

In Ishibashi & Shimodaira (2023), it is argued that unlearning approaches can lead to hallucination and may not make the generation harmless. To address this, they propose a different approach called knowledge sanitization. This approach uses LoRA to instruct the model to respond with a *sanitization phrase* like "I don't know" when encountering potentially harmful prompts. To achieve this, they build a training set where answers to harmful questions are removed. This method, similar to Chen & Yang (2023), is evaluated on both data to be retained and data to be forgotten. The performance is high for both retaining and forgetting targets.

For large language models, In-Context Learning can be used for unlearning, as demonstrated by ICUL (In-Context UnLearning) (Pawelczyk et al., 2023). This technique builds a context where the data point to be unlearned is included but with a flipped label (e.g., changing a positive label to negative) alongside correctly labeled examples. Compared to Jang et al. (2022), this approach achieves similar performance for both unlearning and the primary task without requiring any parameter updates. While this method was tested on a binary sentiment classification task, theoretically, it could be applied to other tasks by incorrectly labeling categorical samples, though no experiments have been conducted in such contexts to our knowledge.

Model Editing can be also used, in principle, to perfom unlearning of private information. Building on the idea that the linear layers in the Transformer architecture can be interpreted as key-value memories that store information (Geva et al., 2021), different model editing techniques have been developed to modify – without retraining – these specific parameters. Meng et al. (2023a) introduce ROME (Rank-One Model Editing), allowing factual predictions to be traced to specific layers for individual manipulation. Furthermore, a similar approach called MEMIT (Meng et al., 2023b) proves to effectively modify multiple examples without affecting the utility of the model. While grey-box attacks that examine intermediate hidden states can still recover deleted data 38% of the time (Patil et al., 2023), and being mostly focused on factual associations, ROME and MEMIT show potential for tracing other data types, substituting the private information with an anonymous one.

With a more specific focus on privacy, Wu et al. (2023b) introduce DEPN (Detecting and Editing Privacy Neurons) inspired by ROME. This method assume that private information, like factual information, might be encoded in specific neurons. They propose detecting these neurons and setting their weights to zero to block information flow. Results show good validation perplexity, sometimes surpassing models trained with Differential Privacy but with higher privacy risks. However, DEPN significantly reduces training time, offering a better balance between computational efficiency and performance. Venditti et al. (2024) focus on the idea of breaking the association between an individual and its private information, inadvertently included into the training data. Their method is called PAE and is based on MEMIT: PAE aims to make multiple edits feasible, both in a sequential and batch fashion, by taking into account the norm of the update matrix with respect to the original weights, without sacrificing model's utility.

5.5 Homomorphic Encryption

Homomorphic Encryption

Definition: Homomorphic Encryption (HE) is a cryptographic technique that allows computations to be performed directly on encrypted data without the need to decrypt it first. The result of these computations, when decrypted, matches the result of operations performed on the plaintext data.

Homomorphic Encryption (HE) offers a theoretical framework for preserving privacy in machine learning by enabling computations on encrypted data. In the context of Large Language Models (LLMs), HE can be considered for performing inference on sensitive inputs without exposing the raw data to the model provider. This could be particularly useful when the LLM is hosted on external servers or cloud services, and users wish to keep their input data confidential. Regarding training, HE is likely to not be able to prevent memorization as the patterns in the plaintext that are usually memorized by LLMs should be preserved by the encryption step.

The application of Homomorphic Encryption to LLMs aims to address privacy concerns during the inference phase by ensuring that sensitive user inputs remain encrypted throughout the computational process. Users encrypt their queries using an HE scheme and send the encrypted data to the server hosting the LLM. The server performs computations directly on the encrypted data and returns an encrypted result, which the user can decrypt to obtain the plaintext output. While this approach theoretically provides strong privacy guarantees, several significant challenges hinder its practical implementation with LLMs.

First, **computational overhead and latency** are substantial. HE schemes, especially those capable of supporting the complex operations required by deep neural networks, introduce significant computational overhead. Operations on encrypted data are significantly slower than their plaintext counterparts, often by several orders of magnitude. This performance penalty makes real-time inference with LLMs impractical using current HE technologies. Second, **model complexity and compatibility** pose challenges. LLMs typically consist of billions of parameters and involve complex architectures with non-linear activation functions, such as GELU or Softmax, and operations like layer normalization and attention mechanisms. Many of these operations are not directly compatible with standard HE schemes, which primarily support arithmetic operations (addition and multiplication). Approximating these non-linear functions with polynomial functions compatible with HE can lead to a loss in model accuracy and additional computational complexity. Additionally, **memory and storage requirements** increase significantly. Encrypted data under HE schemes can be much larger than plaintext data due to encryption overhead. This ciphertext expansion results in higher memory and storage requirements on both client and server sides, making it challenging to handle large encrypted inputs and model parameters. Finally, there is **limited support for training**. While HE can, in theory, be applied to both training and inference, the vast computational demands make training LLMs under HE impractical with current technology. Consequently, most considerations of HE in the context of LLMs focus solely on the inference phase.

Current Research and Limitations Research into applying Homomorphic Encryption to neural networks has shown some success with smaller models and simpler tasks. For instance, Gilad-Bachrach et al. (2016) introduced CryptoNets, demonstrating the feasibility of performing inference on encrypted data with neural networks for basic classification tasks like recognizing handwritten digits (MNIST dataset). Similarly, Badawi et al. (2020) proposed optimizations for running convolutional neural networks under HE but were limited to relatively small models and datasets. Extending these approaches to LLMs presents additional challenges.

Recent efforts have attempted to apply HE to Transformer models and LLMs. Hao et al. (2022) proposed *Iron*, a method that modifies Transformers to be more compatible with HE by approximating non-linear operations with polynomial functions, aiming to reduce computational complexity while preserving accuracy. Pang et al. (2024) introduced *BOLT*, which provides privacy-preserving and efficient inference for Transformers by reformulating attention computation to be more HE-friendly. Similarly, Chen et al. (2022a) presented *THE-X*, a framework that redesigns critical components of the Transformer architecture to suit HE constraints, optimizing encoding schemes to minimize performance degradation. Zheng et al. (2023) developed

Primer, focusing on fast private Transformer inference on encrypted data by introducing optimizations like efficient polynomial approximations and streamlined encrypted computations.

Recently, Hou et al. (2023) introduced *CipherGPT*, a framework designed for secure two-party GPT inference that addresses privacy concerns in LLMs. CipherGPT enables a client (holding the input data) and a server (hosting the model) to collaboratively perform GPT inference without revealing the client's input or the server's model parameters to each other. This is achieved through the use of cryptographic protocols such as secure multiparty computation and homomorphic encryption, optimized specifically for GPT models.

CipherGPT introduces several technical innovations to improve the efficiency and practicality of secure inference with large models like GPT. These include a customized secure matrix multiplication protocol using subfield vector oblivious linear evaluation (sVOLE), an efficient method for computing the GELU activation function using spline-based approximations, and protocols for secure top-K selection and sampling required for the text generation process. By addressing the computational and communication overheads typically associated with homomorphic encryption, CipherGPT demonstrates the feasibility of performing privacy-preserving inference with large-scale language models. However, despite these advancements, CipherGPT still faces challenges related to computational overhead and scalability. The secure protocols, although optimized, introduce latency and resource consumption that may hinder real-time applications, especially with larger models. Additionally, the approximations used for non-linear functions may affect the model's performance and the quality of the generated text. Implementing CipherGPT also requires specialized expertise in cryptography and deep learning, which can be a barrier to widespread adoption.

While these works represent significant progress, they are often constrained by computational overhead and are applied to smaller models or simplified versions of Transformers. One major challenge is **approximating non-linear functions**. LLMs rely heavily on non-linear operations not directly supported by HE schemes. Researchers have explored using polynomial approximations of these functions, but such approximations can degrade model performance. For example, approximating the Softmax function used in attention mechanisms with a polynomial can lead to less accurate attention scores, affecting the overall quality of the generated text.

Another issue is the pronounced **performance trade-off** between privacy and utility. With LLMs, due to their size and complexity, this trade-off is more significant. While HE provides strong privacy by keeping data encrypted, the resulting computational costs and potential loss in accuracy make it less feasible for practical applications involving LLMs.

Furthermore, there is a **lack of practical implementations** for full-scale LLMs. Although recent works like Rho et al. (2024) have provided implementations for encoder-only models, they still face scalability issues when applied to large models. Li et al. (2022) introduced *MPCFormer*, which enables private Transformer inference using Secure Multiparty Computation (MPC) instead of HE, optimizing the Transformer architecture to be compatible with MPC protocols. While MPCFormer shows improved performance over traditional HE methods, it still encounters challenges with communication overhead and scalability when applied to large-scale LLMs.

While Homomorphic Encryption presents an attractive theoretical solution for preserving privacy during inference with Large Language Models, current technological limitations render it impractical for real-world applications involving LLMs. The substantial computational overhead, memory requirements, and challenges in handling complex non-linear operations make HE unsuitable for deployment with current models. Alternative privacy-preserving techniques, such as Secure Multiparty Computation (SMC) and Trusted Execution Environments (TEEs), face similar challenges when applied to LLMs. Differential Privacy (as discussed in §5.2) and Federated Learning (§5.1) currently offer more practical approaches for preserving privacy in the context of LLMs, balancing efficiency with privacy guarantees. Ongoing research in cryptographic techniques and hardware acceleration may, in the future, alleviate some of the computational burdens associated with Homomorphic Encryption. Until such advancements are realized, HE remains a topic of interest in theoretical discussions but is not a practical solution for privacy preservation in Large Language Models.

6 Tools & Frameworks

After discussing the different approaches one may take to tackle the problem of privacy-preserving LLMs, we briefly see what frameworks and libraries are already in place for researchers and practitioners who are interested in the topic.

Data Anonymization. Data anonymization has been extensively used to tackle privacy problems in ML for years, which means several frameworks and libraries can apply anonymization to datasets. The most prominent examples are:

- **ARX Data Anonymization Tool**: ARX is a comprehensive open-source tool that provides scalability and usability for data anonymization. It supports a variety of anonymization techniques, data quality analysis methods, re-identification risk assessments, and privacy models like k-anonymity, l-diversity, t-closeness, and differential privacy. It's updated regularly and is available in Java.
- **Open Data Anonymizer**: Aimed at Python users, this library focuses on data anonymization and masking for data science tasks. It integrates well with Python's data-handling libraries, such as pandas, making it a suitable choice for projects involving data frames and machine learning.
- **Microsoft Presidio**: Presidio offers detailed anonymization capabilities for text, including the identification of personally identifiable information (PII) and its subsequent anonymization or de-identification. It features two main modules: the Presidio Analyzer for detecting PII in text and the Presidio Anonymizer for removing detected PII using various operators like redaction, replacement, hashing, or encryption.
- **Spark NLP for Healthcare** by John Snow Labs: Specifically designed for the healthcare sector, Spark NLP for Healthcare provides state-of-the-art accuracy for extracting, classifying, and structuring clinical and biomedical text data. It's optimized for Databricks, offering solutions like OCR for document processing and NLP models for detecting and obfuscating protected health information (PHI) in unstructured text

Differential Privacy We have seen that Differential Privacy has been considered a gold standard for privacy protection in machine learning for years now, with several techniques to enforce it and big players using it in very used applications. Naturally, this means that there is a high number of frameworks to implement it, some of which are open source. Most of these frameworks, like most of the research in the field, focus on differentially private training. The most important ones are:

- **TensorFlow Privacy**: Similar to Opacus but for TensorFlow, it is a Python library that includes implementations of TensorFlow optimizers for training machine learning models with DP. The library has tutorials and analysis tools for computing the privacy guarantees provided.
- **PyVacy**: PyTorch translation of TensorFlow Privacy.
- **OpenDP project**: incubated by Harvard University's Privacy Tools and Privacy Insights projects, the OpenDP project (OpenDP, 2020) aims to make DP tools available to everyone freely. It includes several components: OpenDP Library, a core component consisting of a collection of algorithms for generating differentially private statistical releases, developed in Rust with Python bindings; SmartNoise, designed for creating DP reports, dashboards, synopses, and synthetic data releases. It supports SQL queries over Spark and popular database engines and includes a variety of synthesizers for generating privacy-preserving synthetic data; DP Creator, a web-based application, is focused on enabling the calculation of DP statistics for statistical queries intended for public release.
- **Diffprivlib**: IBM's general-purpose library (Holohan et al., 2019), stands out for its focus on providing a wide range of DP tools for ML and data analysis tasks, supporting experimentation and development.

- **Google DP**: Provides a broad set of DP tools in Java, Go and C++. (1) Privacy on Beam: an end-to-end DP framework built on top of Apache Beam. (2) Three "DP building block" libraries in C++, Go, and Java implement basic noise addition primitives and DP aggregations. Privacy on Beam is implemented using these libraries. (3) A stochastic tester is used to help catch regressions that could make the DP no longer hold. (4) A DP accounting library, used for tracking privacy budget. (5) A command line interface for running differentially private SQL queries with ZetaSQL. DP Auditorium is a library for auditing differential privacy guarantees.

- **dp-few-shot-generation**: Microsoft's contribution to the DP landscape (Tang et al., 2024) is the only inference-DP framework you will find in this list. As explained in §5.2, DP in-context-learning is a relatively new idea, and this is the only public library to do it, to the best of our knowledge.

- **EKTELO**: made by Zhang et al. (2018), it stands out for its focus on a flexible, extensible framework for privacy-preserving data analysis, enabling complex queries and analytics while ensuring differential privacy. As described on the GitHub page, they have 2 main objectives: Isolate private interactions with data in a compact and secure kernel and modularize privacy-related algorithms into operators, which promote code reuse and assist in keeping the kernel compact and secure.

- **PyTorch Opacus**: made by Yousefpour et al. (2021), it enables training PyTorch models with differential privacy. It supports training with minimal code changes required on the client, has little impact on training performance, and allows the client to track online the privacy budget expended at any given moment. This is all taken care of by an abstraction called PrivacyEngine.

- **private-transformers**: Provides a privacy engine built off Opacus but rewritten specifically to facilitate integration with HuggingFace transformers.

- **dp-transformers**: It is a toolkit that provides a simplified integration of transformers training with DP. It also provides some examples and tips for DP training.

- **Chorus**: Chorus (Johnson et al., 2018) is unique for its DBMS-independent architecture, focusing on scalable, DP statistical queries through a cooperative query processing system. This library was actually deployed at Uber.

- **autodp**: Automates the process of calculating the privacy guarantees for complex algorithms and supports several standard mechanisms. Users can define their own privacy mechanisms or modify existing ones.

Transformers The main focus of this work is LLMs, which are based on the transformers architecture (Vaswani et al., 2023). Most transformers nowadays are implemented with the Transformers Library by HuggingFace. This library quickly became a cornerstone in the field of AI because it provides most state-of-the-art models already pre-trained and ready to be used in a plug-and-play fashion, allowing researchers to fast and efficient prototyping. It provides tools for pre-training and fine-tuning like SFT for supervised learning and TRL for reinforcement learning. The main reason for the success of this library is its versatility: there are also integrations for distributed training with Accelerate and DeepSpeed and it even supports Parameter-Efficient Fine-Tuning with an ad-hoc library called PEFT, which are all fundamental elements when you use big models like modern LLMs. There has been some work to try and integrate DP libraries and Transformers, and in particular there are two public implementations of Opacus with Transformers: private-transformers and dp-transformers which both provide implementations of some recent LLMs and other models with the integration of Opacus for Differential Privacy.

Federated Learning Federated learning is standard when the task requires handling sensitive data, as it makes it possible to keep the data on-site without sending them to a central server and therefore gives a (false §3.3) sense of security. There are several frameworks for federated learning. Let's quickly see the most famous ones:

Framework	Optimizer	Mechanisms	Models	ICL	Language
Tensorflow Privacy	✓	✗	✗	✗	Python
PyVacy	✓	✗	✗	✗	Python
OpenDP	✗	✓	✗	✗	Rust [a]
Diffprivlib (IBM)	✗	✓	✓	✗	Python
Google DP	✗	✓	✗	✗	C++,Java,Go [b]
dp-few-shot-generation	✗	✗	✗	✓	Python
EKTELO	✗	✓	✗	✗	Python
Opacus	✓ [c]	✗	✗	✗	Python
private-transformers	✓ [d]	✗	✗	✗	Python
dp-transformers	✓ [e]	✗	✗	✗	Python
Chorus	✗	✓	✗	✗	Scala
autodp	✗	✓	✗	✗	Python

Table 1: Comparison of Differential Privacy Frameworks. Categorized here based on the components they offer: Optimizer means that the framework offers some optimizer that can be used in combination with any model to train it with DP; Mechanism means that the framework provides noise injection mechanisms like the Exponential Mechanism; Models means that the framework provides some DP models ready to be used; In-Context-Learning means that the framework offers the possibility to perform ICL with your model.

[a] Bindings for Python
[b] Bindings for Python
[c] Opacus provides a privacy engine that encapsulates the model and the optimizer
[d] Built off Opacus, which provides a privacy engine that encapsulates the model and the optimizer
[e] Built off Opacus, which provides a privacy engine that encapsulates the model and the optimizer

- **Flower**: Flower (Beutel et al., 2022) is an open-source,framework-agnostic FL library designed to enable the building of distributed systems with any ML framework. It focuses on usability, scalability, and compatibility, aiming to make FL accessible to researchers and practitioners working with different tools. Flower provides a flexible and easy-to-use solution since it allows for experimentation across different frameworks, making it ideal for researchers and developers exploring federated learning in diverse environments.

- **pfl**: Apple's pfl library is a federated learning library specifically devised for quick research simulations. It also provides tight integration with both local and central DP mechanisms, allowing quick prototyping of differentially private federated networks.

- **TensorFlow Federated (TFF)**: TensorFlow Federated is an open-source framework for machine learning and other computations on decentralized data. TFF enables developers to simulate FL in their machine learning models using TensorFlow. It's designed for flexibility and ease of use when experimenting with FL algorithms. TFF is a natural choice for those who use TensorFlow since it provides seamless integration with existing TensorFlow models and extensive documentation and tutorials from Google.

- **PySyft**: PySyft is a Python library for secure and private deep learning that extends PyTorch. It supports FL, DP, and SMPC, aiming to make privacy-preserving machine learning accessible. PySyft is part of the OpenMined initiative, which fosters an open-source community focusing on privacy-preserving AI.

- **FATE (Federated AI Technology Enabler)**: Webank's FATE is an open-source project intended to provide a secure computing framework to support the federated AI ecosystem. It's designed for financial institutions, enabling them to build machine learning models without sharing their data, thus complying with regulations. FATE is particularly suited for organizations in the financial sector looking to engage in federated learning. It provides a robust framework that emphasizes security and privacy, making it ideal for use cases that require strict data protection measures.

- **PaddleFL**: Baidu's PaddleFL is an open-source federated learning framework based on PaddlePaddle, Baidu's ML platform. It supports various FL strategies and provides a flexible definition of federated learning tasks. It's designed with industrial applications in mind, offering tools for model training and inference in a privacy-preserving manner. This is the go-to for those who are already using PaddlePaddle.

7 Limitations and Future Directions

In this section, we provide a discussion on current limitations and future directions for preserving privacy in large language models, analyzing all the different approaches to preserving privacy proposed in this survey.

Preserving Privacy in Data. Incorporating privacy directly within data before it is input into a large language model offers a practical alternative to more complex, model-based privacy-preserving techniques. For example, the large-scale pretraining of prominent models like LLaMA3 and OLMo leverages data anonymization techniques, applying privacy measures at the data level before it enters the LLM. However, given the vast scale of pretraining data, these projects have largely relied on basic regular expressions to filter out PII and streamline the preprocessing (Dubey et al., 2024; Soldaini et al., 2024). While efficient, this approach is quite limited as it lacks comprehensive privacy protection and does not offer formal privacy guarantees such as those provided by DP. As Subramani et al. (2023) notes, "Running an out-of-the-box tool such as Presidio is the bare minimum that should be applied to all new and existing corpora". To support progress in this area, sections 4.1 and 6 of this survey present practical insights for researchers and practitioners aiming to build the next generation of privacy-preserving LLMs. While current models primarily rely on regex-based filtering for pretraining data, several approaches discussed in section 4.1, including metric Differential Privacy methods, can be scaled and applied to enhance privacy preservation in LLMs. We believe these methods hold significant untapped potential for advancing privacy-preserving techniques in data handling for LLMs, and that we are just beginning to explore their full possibilities.

Preserving Privacy in Model. Model-based approaches based on DP offer mathematically-proven privacy guarantees within defined privacy parameters (ϵ, δ), but they come with significant computational costs and require substantial, often complex, modifications to models and training procedures. Of all the approaches discussed in section 5, DP is considered the gold standard for privacy-preserving machine learning, as it is the only method that provides a formal mathematical privacy guarantee. However, this assurance typically comes at the expense of some utility loss, especially when DP is applied throughout the entire learning pipeline (Tramèr et al., 2024; Feng & Tramèr, 2024). DP's applicability spans data preparation, training, fine-tuning, and inference. In contrast, the privacy guarantee provided by other approaches like Machine Unlearning (§5.4) is only empirical but necessary to implement the "right to be forgotten" of privacy laws like GDPR (Zhang et al., 2024b). Future research could focus on building a rigorous mathematical basis for Machine Unlearning, developing techniques to seamlessly combine it with DP, and implementing accessible interfaces that allow users to specify data they wish to be erased from the model.

Despite the extensive application scope of Differential Privacy, it assumes a structure of private data where every bit of language can potentially contain secrets that are not meant to be shared indiscriminately. Differential Privacy cannot directly preserve secrets – information that can be shared only to the correct group of people (Brown et al., 2022b). The capability of LLMs to keep secrets is still little explored, with the development of the first benchmarks to assess the ability of such models to disclose private information only to users who should be allowed to access them (Mireshghallah et al., 2023; Priyanshu et al., 2023; Wang et al., 2024a). Future research should focus on advancing DP to selectively protect specific language elements, enhancing utility by applying privacy constraints only where necessary. As LLMs are increasingly integrated as assistants in applications, functioning as agents among groups of users who wish to share certain information while keeping other details private, addressing this issue will become more critical.

Looking ahead, applying DP to the training, tuning, or conditioning of larger, more capable models presents both a challenge and an opportunity. Although most current studies focus on smaller models, the advancement of quantization and Parameter Efficient Fine-Tuning (PEFT) techniques promises to make DP viable in larger models as well. These advancements reduce computational overhead and have facilitated DP-tuning

in smaller models, such as RoBERTa and GPT-2 via LoRA, which could soon extend to modern models like Llama-3 and Mistral with negligible impact on computational resources (§5.1.3). We believe that the next step in this field will be the development of privacy-preserving LLMs that leverage these techniques, paired with state-of-the-art models. With the existing building blocks (techniques and models) in place, this development hinges primarily on computational resources for private fine-tuning. Our ideal vision includes models pre-trained entirely with DP, which would significantly mitigate privacy risks from the outset. However, we recognize that fully DP-compliant pre-training remains challenging, as it demands significant computational resources and can reduce model effectiveness. A practical compromise could involve pre-training the model using data anonymization techniques, which are less computationally intensive (as discussed in Section 4.1), while reserving DP methods for the subsequent stages of instruction tuning and alignment. In the short term, we believe this approach offers a reasonable balance.

Federated Learning (§5.3) offers another promising avenue for enhancing privacy in LLMs, albeit with limitations. Although FL with Central DP is feasible, constraints on device capacity mean that edge devices, such as mobile phones, often lack the processing power to maintain a locally stored LLM. This limitation is slowly diminishing, as illustrated by devices capable of running small-scale models like Gemini-Nano locally (Team et al., 2024). With the ongoing miniaturization and optimization of LLMs, we anticipate that FL-driven, DP-compliant training on local devices will become increasingly feasible. Nevertheless, implementing a local DP model remains challenging due to the high dimensionality of the data and model size; PEFT techniques provide some relief by minimizing parameter sharing, but more research is needed to refine utility and privacy outcomes within this framework.

Inference under DP constraints has shown initial success, suggesting that LLMs may offer privacy-preserving inference without the need for task-specific fine-tuning (§5.2). While current research in DP-constrained inference is limited, early results are promising, especially as models become more compact without sacrificing capability. In practical terms, this progress allows for privacy-preserving inference in cases where offsite models are preferable or where specific fine-tuning is unnecessary. Future work could explore larger models, potentially introducing DP to inference on locally stored models, where the option of performing inference without any data sharing would be highly advantageous.

Despite the appeal of Homomorphic Encryption (HE) for privacy-preserving inference with LLMs, significant challenges limit its practicality. As outlined in §5.5, HE introduces considerable computational overhead and latency, making real-time inference with large models unfeasible. Issues with approximating non-linear functions can degrade performance, while ciphertext expansion adds memory demands that complicate deployment on limited-resource systems. Recent advancements like CipherGPT (Hou et al., 2023) have optimized HE for LLMs, yet scalability, performance trade-offs, and complexity remain concerns. Future research should aim to reduce overhead through more efficient protocols, improve non-linear approximations, and explore hybrid methods that combine HE with other privacy techniques.

In conclusion, our findings reveal that while privacy-preserving techniques for ML are advancing, efficiency remains a central focus, as existing methods often involve trade-offs in training speed and model performance. Recent innovations have brought privacy-preserving approaches closer to practical deployment in public-facing models, reducing these trade-offs and making DP increasingly accessible. We are confident that this trend will persist, paving the way for a widespread use of privacy-preserving ML as a practical alternative in the near future.

8 Conclusion

In this work, we conducted an extensive survey on preserving privacy in Large Language Models (LLMs), meticulously analyzing the various dimensions of privacy concerns and the corresponding solutions.

Firstly, our investigation began by exploring the different types of privacy attacks (§3) that these models are susceptible to, including Training Data Extraction (§3.1), Membership Inference (§3.2), and Model Inversion (§3.3). Each type exposes significant vulnerabilities. Training Data Extraction attacks compromise privacy by enabling adversaries to reconstruct sensitive training data such as personal identifiers or confidential communications, which erodes trust in LLM applications. Membership Inference attacks reveal whether

specific data was used in training, potentially disclosing sensitive personal details like medical conditions, which could lead to discrimination. Model Inversion further extends these risks by allowing attackers to deduce personal characteristics from model outputs, promoting unauthorized profiling and significant privacy breaches.

Secondly, we presented an extensive view of the state-of-the-art solutions for preserving privacy in large language models (§4-5). These solutions include approaches that inject privacy directly in the training data like anonymization techniques (§4) or approaches that act on the model (§5). While there are many possible approaches to the problem, no single solution fits all scenarios. We presented solutions for applying privacy mechanisms at several points in the learning pipeline: from data preparation (before training - §4), to private training (during training - §5.1), to private inference (§5.2-5.5), and finally after training (§5.4). We refer the reader to section 7 for a thorough discussion on current limitations and future directions for preserving privacy in large language models.

This work highlights the critical need to address privacy vulnerabilities in Large Language Models (LLMs) by systematically analyzing various privacy attacks and their profound risks to data confidentiality and user trust. We presented current state-of-the-art solutions such as data anonymization, Differential Privacy, and Machine Unlearning, emphasizing the importance of continuous advancement and integration of these techniques. As LLMs become integral to sensitive domains like healthcare and finance, developing robust privacy safeguards is essential. Future research must enhance these methods, improve the privacy-utility trade-off, and ensure that larger models adhere to stringent privacy standards, ultimately fostering the creation of secure and reliable LLM technologies that uphold user privacy and trust.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16. ACM, October 2016. doi: 10.1145/2976749.2978318. URL http://dx.doi.org/10.1145/2976749.2978318.

Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun hsuan Sung, Brian Strope, and Ray Kurzweil. Conversational contextual cues: The case of personalization and history for response ranking, 2016.

Galen Andrew, Om Thakkar, H. Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping, 2022.

Avi Arampatzis, George Drosatos, and Pavlos S. Efraimidis. Versatile query scrambling for private web search. *Inf. Retr.*, 18(4):331–358, aug 2015. ISSN 1386-4564. doi: 10.1007/s10791-015-9256-0. URL https://doi.org/10.1007/s10791-015-9256-0.

Stefan Arnold, Dilara Yesilbas, and Sven Weinzierl. Guiding text-to-text privatization by syntax, 2023.

Ahmad Al Badawi, Jin Chao, Jie Lin, Chan Fook Mun, Jun Jie Sim, Benjamin Hong Meng Tan, Xiao Nan, Khin Mi Mi Aung, and Vijay Ramaseshan Chandrasekhar. Towards the alexnet moment for homomorphic encryption: Hcnn, thefirst homomorphic cnn on encrypted data with gpus, 2020. URL https://arxiv.org/abs/1811.00778.

Mislav Balunovic, Dimitar Dimitrov, Nikola Jovanović, and Martin Vechev. Lamp: Extracting text from gradients with language model priors. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 7641–7654. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/32375260090404f907ceae19f3564a7e-Paper-Conference.pdf.

Samah Baraheem and Zhongmei Yao. A survey on differential privacy with machine learning and future outlook. *arXiv preprint arXiv:2211.10708*, 2022.

Priya Basu, Tiasa Singha Roy, Rakshit Naidu, Zumrut Muftuoglu, Sahib Singh, and Fatemehsadat Mireshghallah. Benchmarking differential privacy and federated learning for bert models. *ArXiv*, abs/2106.13973, 2021. URL https://api.semanticscholar.org/CorpusID:235658799.

Björn Bebensee. Local differential privacy: a tutorial, 2019.

Rouzbeh Behnia, Mohammadreza Reza Ebrahimi, Jason Pacheco, and Balaji Padmanabhan. Ew-tune: A framework for privately fine-tuning large language models with differential privacy. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, November 2022. doi: 10.1109/icdmw58026.2022.00078. URL http://dx.doi.org/10.1109/ICDMW58026.2022.00078.

Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane. Flower: A friendly federated learning research framework, 2022.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. ISBN 978-0387310732.

G. R. Blakley. Safeguarding cryptographic keys. *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pp. 313–318, 1899. URL https://api.semanticscholar.org/CorpusID:38199738.

Franziska Boenisch, Christopher Muhl, Roy Rinberg, Jannis Ihrig, and Adam Dziedzic. Individualized pate: Differentially private machine learning with individual privacy guarantees. *Proc. Priv. Enhancing Technol.*, 2023:158–176, 2022. URL https://api.semanticscholar.org/CorpusID:252693510.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pp. 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3133982. URL https://doi.org/10.1145/3133956.3133982.

Amine Boulemtafes, Abdelouahid Derhab, and Yacine Challal. A review of privacy-preserving techniques for deep learning. *Neurocomputing*, 384:21–45, 2020. URL https://api.semanticscholar.org/CorpusID:211159333.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, 2019. URL https://api.semanticscholar.org/CorpusID:208909851.

Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pp. 234–238. Springer, 1986. doi: 10.1007/3-540-47721-7_17.

Alissa Brauneck, Louisa Schmalhorst, Mohammad Mahdi Kazemi Majdabadi, Mohammad Bakhtiari, Uwe Völker, Jan Baumbach, Linda Baumbach, and Gabriele Buchholtz. Federated machine learning, privacy-enhancing technologies, and data protection laws in medical research: Scoping review. *Journal of Medical Internet Research*, 25, 2023. URL https://api.semanticscholar.org/CorpusID:257834709.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 2280–2292, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3534642. URL https://doi.org/10.1145/3531146.3534642.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2280–2292, 2022b.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. *ArXiv*, abs/2206.07136, 2022a. URL https://api.semanticscholar.org/CorpusID:249674655.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Differentially private optimization on large model at small cost. *ArXiv*, abs/2210.00038, 2022b. URL https://api.semanticscholar.org/CorpusID:252683151.

Zhiqi Bu, Ruixuan Liu, Yu xiang Wang, Sheng Zha, and George Karypis. On the accuracy and efficiency of group-wise clipping in differentially private optimization. *ArXiv*, abs/2310.19215, 2023. URL https://api.semanticscholar.org/CorpusID:264818104.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. *2015 IEEE Symposium on Security and Privacy*, pp. 463–480, 2015. URL https://api.semanticscholar.org/CorpusID:5945696.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks, 2019.

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

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023a.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, et al. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*, 2024.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023b.

Aldo Gael Carranza, Rezsa Farahani, Natalia Ponomareva, Alex Kurakin, Matthew Jagielski, and Milad Nasr. Synthetic query generation for privacy-preserving deep retrieval systems using differentially private language models, 2024.

Ricardo Silva Carvalho, Theodore Vasiloudis, and Oluwaseyi Feyisetan. Tem: High utility metric differential privacy on text. *ArXiv*, abs/2107.07928, 2021. URL https://api.semanticscholar.org/CorpusID:236034456.

Andre Chailloux, Gus Gutoski, and Jamie Sikora. Optimal bounds for semi-honest quantum oblivious transfer. *Chic. J. Theor. Comput. Sci.*, 2016, 2013. URL https://api.semanticscholar.org/CorpusID:2367049.

Tyler A. Chang and Benjamin K. Bergen. Language model behavior: A comprehensive survey. *ArXiv*, abs/2303.11504, 2023. URL https://api.semanticscholar.org/CorpusID:257636789.

Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *International Symposium on Privacy Enhancing Technologies*, 2013. URL https://api.semanticscholar.org/CorpusID:14750765.

Dingfan Chen. A unified view of differentially private deep generative modeling. *ArXiv*, abs/2309.15696, 2023. URL https://api.semanticscholar.org/CorpusID:262940016.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms, 2023.

Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.findings-acl.355. URL http://dx.doi.org/10.18653/v1/2023.findings-acl.355.

Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. The-x: Privacy-preserving transformer inference with homomorphic encryption. In *Findings*, 2022a. URL https://api.semanticscholar.org/CorpusID:248780015.

Wei-Ning Chen, Ayfer Ozgur, and Peter Kairouz. The poisson binomial mechanism for unbiased federated learning with secure aggregation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3490–3506. PMLR, 17–23 Jul 2022b. URL https://proceedings.mlr.press/v162/chen22s.html.

Yiyi Chen, Heather Lent, and Johannes Bjerva. Text embedding inversion security for multilingual language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7808–7827, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.422. URL https://aclanthology.org/2024.acl-long.422.

Yu Chen, Tingxin Li, Huiming Liu, and Yang Yu. Hide and seek (has): A lightweight framework for prompt privacy protection. *ArXiv*, abs/2309.03057, 2023b. URL https://api.semanticscholar.org/CorpusID:261557670.

Kai Yuen Cheong, Takeshi Koshiba, and Shohei Nishiyama. Strengthening the security of distributed oblivious transfer. In *Australasian Conference on Information Security and Privacy*, 2009. URL https://api.semanticscholar.org/CorpusID:41788784.

Junjie Chu, Zeyang Sha, Michael Backes, and Yang Zhang. Reconstruct your previous conversations! comprehensively investigating privacy leakage risks in conversations with gpt models, 2024. URL https://arxiv.org/abs/2402.02987.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. On the use of arxiv as a dataset, 2019.

Cameron Coles. 11URL https://www.cyberhaven.com/blog/4-2-of-workers-have-pasted-company-data-into-chatgpt.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *ArXiv*, abs/2204.13650, 2022. URL https://api.semanticscholar.org/CorpusID:248427073.

Lea Demelius, Roman Kern, and Andreas Trügler. Recent advances of differential privacy in centralized deep learning: A systematic survey. *ArXiv*, abs/2309.16398, 2023. URL https://api.semanticscholar.org/CorpusID:263134574.

Jieren Deng, Yijue Wang, Ji Li, Chenghong Wang, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. TAG: Gradient attack on transformer-based language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3600–3610, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.305. URL https://aclanthology.org/2021.findings-emnlp.305.

Erik Derner, Kristina Batistic, Jan Zah'alka, and Robert Babuka. A security risk taxonomy for large language models. *ArXiv*, abs/2311.11415, 2023. URL https://api.semanticscholar.org/CorpusID:265294893.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:3277268.

Youlong Ding, Xueyang Wu, Yining Meng, Yonggang Luo, Hao Wang, and Weike Pan. Delving into differentially private transformer, 2024.

My-Hoa Nathalie Dinh and Ferdinando Fioretto. Context-aware differential privacy for language modeling. *ArXiv*, abs/2301.12288, 2023. URL https://api.semanticscholar.org/CorpusID:256389732.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *ArXiv*, abs/2210.09929, 2022. URL https://api.semanticscholar.org/CorpusID:252968205.

C.M. Downey, Wei Dai, Huseyin A. Inan, Kim Laine, Saurabh Naik, and Tomasz Religa. Planting and mitigating memorized content in predictive-text language models. *ArXiv*, abs/2212.08619, 2022. URL https://api.semanticscholar.org/CorpusID:254823223.

Minxin Du, Xiang Yue, Sherman S. M. Chow, and Huan Sun. Sanitizing sentence embeddings (and labels) for local differential privacy. In *Proceedings of the ACM Web Conference 2023*, WWW '23, pp. 2349–2359, New York, NY, USA, 2023a. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583512. URL https://doi.org/10.1145/3543507.3583512.

Minxin Du, Xiang Yue, Sherman S. M. Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23. ACM, November 2023b. doi: 10.1145/3576915.3616592. URL http://dx.doi.org/10.1145/3576915.3616592.

Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. On the privacy risk of in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *Advances in Neural Information Processing Systems*, 36, 2024a.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models?, 2024b. URL https://arxiv.org/abs/2402.07841.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Christophe Dupuy, Radhika Arava, Rahul Gupta, and Anna Rumshisky. An efficient dp-sgd mechanism for large scale nlu models. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4118–4122, 2021. URL https://api.semanticscholar.org/CorpusID:236635047.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology (EUROCRYPT 2006)*, volume 4004 of *Lecture Notes in Computer Science*, pp. 486–503. Springer Verlag, May 2006a. URL https://www.microsoft.com/en-us/research/publication/our-data-ourselves-privacy-via-distributed-noise-generation/.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006b. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *ArXiv*, abs/2310.02238, 2023. URL https://api.semanticscholar.org/CorpusID:263608437.

Adel Elmahdy, Huseyin A. Inan, and Robert Sim. Privacy leakage in text classification a data extraction approach. In Oluwaseyi Feyisetan, Sepideh Ghanavati, Patricia Thaine, Ivan Habernal, and Fatemehsadat Mireshghallah (eds.), *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing*, pp. 13–20, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.privatenlp-1.3. URL https://aclanthology.org/2022.privatenlp-1.3.

David Enthoven and Zaid Al-Ars. An overview of federated deep learning privacy attacks and defensive strategies. *ArXiv*, abs/2004.04676, 2020. URL https://api.semanticscholar.org/CorpusID:215548987.

Úlfar Erlingsson, Aleksandra Korolova, and Vasyl Pihur. Rappor: Randomized aggregatable privacy-preserving ordinal response. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014. URL https://api.semanticscholar.org/CorpusID:6855746.

Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, June 1985. ISSN 0001-0782. doi: 10.1145/3812.3818. URL https://doi.org/10.1145/3812.3818.

Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1:77–94, 1988. URL https://api.semanticscholar.org/CorpusID:2950602.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pp. 954–959, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794. doi: 10.1145/3357713.3384290. URL https://doi.org/10.1145/3357713.3384290.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: discovering the long tail via influence estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Shanglun Feng and Florian Tramèr. Privacy backdoors: Stealing data with corrupted pretrained models, 2024. URL https://arxiv.org/abs/2404.00473.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2019a. URL https://api.semanticscholar.org/CorpusID:204800512.

Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. Leveraging hierarchical representations for preserving privacy and utility in text. *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 210–219, 2019b. URL https://api.semanticscholar.org/CorpusID:204801187.

Matthew Finlayson, Xiang Ren, and Swabha Swayamdipta. Logits of api-protected llms leak proprietary information, 2024. URL https://arxiv.org/abs/2403.09539.

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

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, pp. 17–32, USA, 2014. USENIX Association. ISBN 9781931971157.

Arun Ganesh, Mahdi Haghifam, Milad Nasr, Sewoong Oh, Thomas Steinke, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Why is public pretraining necessary for private model training? In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:257038349.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

Gonzalo Munilla Garrido, Xiaoyuan Liu, Florian Matthes, and Dawn Xiaodong Song. Lessons learned: Surveying the practicality of differential privacy in the industry. *ArXiv*, abs/2211.03898, 2022. URL https://api.semanticscholar.org/CorpusID:253398057.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to bgreak privacy in federated learning? *ArXiv*, abs/2003.14053, 2020. URL https://api.semanticscholar.org/CorpusID:214728347.

General Data Protection Regulation. Regulation (eu) 2016/679 of the european parliament and of the council, 4 2016. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679. Article 4(1): Definition of 'personal data'.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Symposium on the Theory of Computing*, 2009. URL https://api.semanticscholar.org/CorpusID:947660.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L. Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images. *ArXiv*, abs/2302.13861, 2023. URL https://api.semanticscholar.org/CorpusID:257219853.

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 201–210, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gilad-bachrach16.html.

Jennifer Gillenwater, Matthew Joseph, Andrés Muñoz Medina, and Mónica Ribero. A joint exponential mechanism for differentially private top-k, 2022.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7:1–32, 1994. URL https://api.semanticscholar.org/CorpusID:6934049.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016. ISBN 978-0262035613.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

Alejandro Guerra-Manzanares, L. Julian Lechuga Lopez, Michail Maniatakos, and Farah E. Shamout. *Privacy-Preserving Machine Learning for Healthcare: Open Challenges and Future Perspectives*, pp. 25–40. Springer Nature Switzerland, 2023. ISBN 9783031395390. doi: 10.1007/978-3-031-39539-0_3. URL http://dx.doi.org/10.1007/978-3-031-39539-0_3.

Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. Recovering private text in federated learning of language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 8130–8143. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/35b5c175e139bff5f22a5361270fce87-Paper-Conference.pdf.

Ivan Habernal. When differential privacy meets nlp: The devil is in the detail. In *Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://api.semanticscholar.org/CorpusID:237431523.

Muhammad Usman Hadi, al tashi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, Qasem Al-Tashi, Amgad Muneer, Mohammed Ali Al-garadi, Gru Cnn, and T5 RoBERTa. Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. 2023. URL https://api.semanticscholar.org/CorpusID:266378240.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private inference on transformers. In *Neural Information Processing Systems*, 2022. URL https://api.semanticscholar.org/CorpusID:255149865.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *ArXiv*, abs/1811.03604, 2018. URL https://api.semanticscholar.org/CorpusID:53207681.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 61–70, 2010. doi: 10.1109/FOCS.2010.85.

Jahid Hasan. Security and privacy issues of federated learning. *ArXiv*, abs/2307.12181, 2023. URL https://api.semanticscholar.org/CorpusID:260125925.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend, 2015.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63, 2020. doi: 10.1162/tacl_a_00299. URL https://aclanthology.org/2020.tacl-1.4.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: The ibm differential privacy library. *ArXiv*, abs/1907.02444, 2019. URL https://api.semanticscholar.org/CorpusID:195798910.

Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng Li, Bo Li, and Zhangyang Wang. Dp-opt: Make large language model your privacy-preserving prompt engineer. *ArXiv*, abs/2312.03724, 2023. URL https://api.semanticscholar.org/CorpusID:266051675.

Shlomo Hoory, Amir Feder, Avichai Tendler, Alon Cohen, Sofia Erell, Itay Laish, Hootan Nakhost, Uri Stemmer, Ayelet Benjamini, Avinatan Hassidim, and Yossi Matias. Learning and evaluating a differentially private pre-trained language model. In *PRIVATENLP*, 2021. URL https://api.semanticscholar.org/CorpusID:235097653.

Xiaoyang Hou, Jian Liu, Jingyu Li, Yu hai Li, and Cheng Hong. Ciphergpt: Secure two-party gpt inference. *IACR Cryptol. ePrint Arch.*, 2023:1147, 2023. URL https://api.semanticscholar.org/CorpusID:260278194.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *ArXiv*, abs/1902.00751, 2019. URL https://api.semanticscholar.org/CorpusID:59599816.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. User review sites as a resource for large-scale sociolinguistic studies. *Proceedings of the 24th International Conference on World Wide Web*, 2015. URL https://api.semanticscholar.org/CorpusID:4498466.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL https://api.semanticscholar.org/CorpusID:235458009.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2038–2047, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.148. URL https://aclanthology.org/2022.findings-emnlp.148.

Timour Igamberdiev and Ivan Habernal. Dp-bart for privatized text rewriting under local differential privacy. *ArXiv*, abs/2302.07636, 2023. URL https://api.semanticscholar.org/CorpusID:256868867.

Timour Igamberdiev, Doan Nam Long Vu, Felix Künnecke, Zhuo Yu, Jannik Holmer, and Ivan Habernal. Dp-nmt: Scalable differentially-private machine translation. *ArXiv*, abs/2311.14465, 2023. URL https://api.semanticscholar.org/CorpusID:265445192.

Yoichi Ishibashi and Hidetoshi Shimodaira. Knowledge sanitization of large language models. *ArXiv*, abs/2309.11852, 2023. URL https://api.semanticscholar.org/CorpusID:262083936.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:252693065.

Marija Jegorova, Chaitanya Kaul, Charlie Mayor, Alison Q. O'Neil, Alexander Weir, Roderick Murray-Smith, and Sotirios A. Tsaftaris. Survey: Leakage and privacy at inference time, 2022.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):160035, May 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.35. URL https://doi.org/10.1038/sdata.2016.35.

Noah M. Johnson, Joseph P. Near, Joseph M. Hellerstein, and Dawn Xiaodong Song. Chorus: a programming framework for building scalable differential privacy mechanisms. *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 535–551, 2018. URL https://api.semanticscholar.org/CorpusID:226266222.

Chihong Joo and Aaram Yun. Homomorphic authenticated encryption secure against chosen-ciphertext attack. *IACR Cryptol. ePrint Arch.*, 2013:726, 2014. URL https://api.semanticscholar.org/CorpusID:17850861.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy, 2015.

Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. *ArXiv*, abs/2103.00039, 2021a. URL https://api.semanticscholar.org/CorpusID:232075850.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2021b.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7403–7412, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.458. URL https://aclanthology.org/2023.emnlp-main.458.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pp. 531–540, 2008. URL https://api.semanticscholar.org/CorpusID:1935.

Aly M. Kassem, Omar Mahmoud, and Sherif Saad. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:266164054.

Gavin Kerrigan, Dylan Slack, and Jens Tuyls. Differentially private language models benefit from public pre-training. *ArXiv*, abs/2009.05886, 2020. URL https://api.semanticscholar.org/CorpusID:221655783.

Eugene Kim. Amazon warns employees not to share confidential information with ChatGPT after seeing cases where its answer 'closely matches existing material' from inside the company, 1 2023. URL https://www.businessinsider.com/amazon-chatgpt-openai-warns-employees-not-share-confidential-information-microsoft-2023-1?international=true&r=US&IR=T.

Jeongsu Kim and Aaram Yun. Secure fully homomorphic authenticated encryption. *IEEE Access*, 9:107279–107297, 2021. URL https://api.semanticscholar.org/CorpusID:236939797.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sung-Hoon Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models. *ArXiv*, abs/2307.01881, 2023. URL https://api.semanticscholar.org/CorpusID:259342279.

Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi (eds.), *Machine Learning: ECML 2004*, pp. 217–226, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-30115-8.

Oleksandra Klymenko, Stephen Meisenbacher, and Florian Matthes. Differential privacy in natural language processing the story so far. In Oluwaseyi Feyisetan, Sepideh Ghanavati, Patricia Thaine, Ivan Habernal, and Fatemehsadat Mireshghallah (eds.), *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing*, pp. 1–11, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.privatenlp-1.1. URL https://aclanthology.org/2022.privatenlp-1.1.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:13193974.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*, 2019.

Satyapriya Krishna, Rahul Gupta, and Christophe Dupuy. Adept: Auto-encoder based differentially private text transformation. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2021. URL https://api.semanticscholar.org/CorpusID:231750016.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. Harnessing large-language models to generate private synthetic text, 2024.

Iyadh Ben Cheikh Larbi, Aljoscha Burchardt, and Roland Roller. Which anonymization technique is best for which nlp task? - it depends. a systematic study on clinical text processing. *ArXiv*, abs/2209.00262, 2022. URL https://api.semanticscholar.org/CorpusID:251979395.

Jaewoo Lee and Daniel Kifer. Scaling up differentially private deep learning with fast per-example gradient clipping. *Proceedings on Privacy Enhancing Technologies*, 2021:128 – 144, 2020. URL https://api.semanticscholar.org/CorpusID:221516787.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better, 2022.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P. Xing, and Haotong Zhang. Mpcformer: fast, performant and private transformer inference with mpc. *ArXiv*, abs/2211.01452, 2022. URL https://api.semanticscholar.org/CorpusID:253264964.

Haoran Li, Yulin Chen, Jinglong Luo, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, and Yangqiu Song. Privacy in large language models: Attacks, defenses and future directions, 2023a.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on ChatGPT. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 4138–4153, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.272. URL https://aclanthology.org/2023.findings-emnlp.272.

Haoran Li, Mingshi Xu, and Yangqiu Song. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 14022–14040, Toronto, Canada, July 2023c. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.881. URL https://aclanthology.org/2023.findings-acl.881.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, abs/2101.00190, 2021. URL https://api.semanticscholar.org/CorpusID:230433941.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori B. Hashimoto. Large language models can be strong differentially private learners. *ArXiv*, abs/2110.05679, 2021. URL https://api.semanticscholar.org/CorpusID:238634219.

Yansong Li, Zhixing Tan, and Yang Liu. Privacy-preserving prompt tuning for large language model services. *ArXiv*, abs/2305.06212, 2023d. URL https://api.semanticscholar.org/CorpusID:258588141.

Nikhil Dandekar Lili Jiang, Meg Risdal. Quora question pairs, 2017. URL https://kaggle.com/competitions/quora-question-pairs.

Peng Liu, Xiangru Xu, and Wen Wang. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity*, 5, 2022. URL https://api.semanticscholar.org/CorpusID:246447197.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models, 2024a.

Xiao-Yang Liu, Rongyi Zhu, Daochen Zha, Jiechao Gao, Shan Zhong, Matt White, and Meikang Qiu. Differentially private low-rank adaptation of large language model using federated learning. *ACM Trans. Manage. Inf. Syst.*, August 2024b. ISSN 2158-656X. doi: 10.1145/3682068. URL https://doi.org/10.1145/3682068. Just Accepted.

Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V. Vasilakos. Privacy and security issues in deep learning: A survey. *IEEE Access*, 9:4566–4593, 2021. URL https://api.semanticscholar.org/CorpusID:231423554.

Natasha Lomas. Italy orders ChatGPT blocked citing data protection concerns | TechCrunch — techcrunch.com. https://techcrunch.com/2023/03/31/chatgpt-blocked-italy/, 2023.

Nils Lukas, A. Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-B'eguelin. Analyzing leakage of personally identifiable information in language models. *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 346–363, 2023. URL https://api.semanticscholar.org/CorpusID:256459554.

Chuan Ma, Jun Li, Kang Wei, Bo Liu, Ming Ding, Long Yuan, Zhu Han, and Harold Vincent Poor. Trusted ai in multiagent systems: An overview of privacy and security for distributed learning. *Proceedings of the IEEE*, 111:1097–1132, 2022. URL https://api.semanticscholar.org/CorpusID:246996518.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers, 2021.

Peihua Mai, Ran Yan, Zhe Huang, Youjia Yang, and Yan Pang. Split-and-denoise: Protect large language model inference with local differential privacy. *ArXiv*, abs/2310.09130, 2023. URL https://api.semanticscholar.org/CorpusID:264127881.

Nuno J. Mamede, J. Baptista, and Francisco Dias. Automated anonymization of text documents. *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1287–1294, 2016. URL https://api.semanticscholar.org/CorpusID:7050746.

Rahat Masood, Dinusha Vatsalan, Muhammad Ikram, and Mohamed Ali Kaafar. Incognito: A method for obfuscating web data. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pp. 267–276, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi: 10.1145/3178876.3186093. URL https://doi.org/10.1145/3178876.3186093.

Justus Mattern, Zhijing Jin, Benjamin Weggenmann, Bernhard Schoelkopf, and Mrinmaya Sachan. Differentially private language models for secure data sharing. In *Conference on Empirical Methods in Natural Language Processing*, 2022a. URL https://api.semanticscholar.org/CorpusID:253107364.

Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. The limits of word level differential privacy. In *NAACL-HLT*, 2022b. URL https://api.semanticscholar.org/CorpusID:248512848.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL https://aclanthology.org/2023.findings-acl.719.

R. Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. How much do language models copy from their training data? evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670, 2023. doi: 10.1162/tacl_a_00567. URL https://aclanthology.org/2023.tacl-1.38.

H. B. McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *ArXiv*, abs/1602.05629, 2016. URL https://api.semanticscholar.org/CorpusID:16861557.

H. B. McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private language models without losing accuracy. *ArXiv*, abs/1710.06963, 2017. URL https://api.semanticscholar.org/CorpusID:195346432.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models, 2018.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023.

Frank D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, pp. 19–30, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585512. doi: 10.1145/1559845.1559850. URL https://doi.org/10.1145/1559845.1559850.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023a.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Stephane Meystre, F Friedlin, Brett South, Shuying Shen, and Matthew Samore. Automatic de-identification of textual documents in the electronic health record: A review of recent research. *BMC medical research methodology*, 10:70, 08 2010. doi: 10.1186/1471-2288-10-70.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8332–8347, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.570. URL https://aclanthology.org/2022.emnlp-main.570.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory, 2023.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, August 2017. doi: 10.1109/csf.2017.11. URL http://dx.doi.org/10.1109/CSF.2017.11.

Fatemehsadat Mirshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmaeilzadeh. Privacy in deep learning: A survey. *ArXiv*, abs/2004.12254, 2020. URL https://api.semanticscholar.org/CorpusID:216553034.

Robin Mitchell. Samsung Fab Data Leak: How ChatGPT Exposed Sensitive Information — electropages.com. https://www.electropages.com/blog/2023/04/how-chatgpt-exposed-sensitive-information, 4 2023.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. Text embeddings reveal (almost) as much as text. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12448–12460, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.765. URL https://aclanthology.org/2023.emnlp-main.765.

John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. Language model inversion, 2023b. URL https://arxiv.org/abs/2311.13647.

Ahmadreza Mosallanezhad, Ghazaleh Beigi, and Huan Liu. Deep reinforcement learning-based text anonymization against private-attribute inference. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://api.semanticscholar.org/CorpusID:202771304.

Sasi Kumar Murakonda and R. Shokri. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *ArXiv*, abs/2007.09339, 2020. URL https://api.semanticscholar.org/CorpusID:220647336.

Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. *Proceedings 2022 Network and Distributed System Security Symposium*, 2020. URL https://api.semanticscholar.org/CorpusID:239998048.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

National Institute of Standards and Technology. Nist special publication 800-63-3: Digital identity guidelines. Special Publication 800-63-3, National Institute of Standards and Technology, 2017. URL https://pages.nist.gov/800-63-3/. Definition of Personally Identifiable Information (PII).

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

Helen Nissenbaum. Privacy as contextual integrity. *Wash. L. Rev.*, 79:119, 2004.

Helen Nissenbaum. Privacy in context: Technology, policy, and the integrity of social life. *Bibliovault OAI Repository, the University of Chicago Press*, 01 2010.

David Odera. Federated learning and differential privacy in clinical health: Extensive survey. *World Journal of Advanced Engineering Technology and Sciences*, 2023. URL https://api.semanticscholar.org/CorpusID:258166346.

OpenAI. March 20 chatgpt outage: Here's what happened: an update on our findings, the actions we've taken, and technical details of the bug, 3 2023. URL https://openai.com/blog/march-20-chatgpt-outage.

Team OpenDP. The opendp white paper. 2020. URL https://projects.iq.harvard.edu/files/opendp/files/opendp_white_paper_11may2020.pdf.

Ahmed El Ouadrhiri and Ahmed M Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE Access*, 10:22359–22380, 2022. URL https://api.semanticscholar.org/CorpusID:246890627.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Mustafa Ozdayi, Charith Peris, Jack FitzGerald, Christophe Dupuy, Jimit Majmudar, Haidar Khan, Rahil Parikh, and Rahul Gupta. Controlling the extraction of memorized data from large language models via prompt-tuning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1512–1521, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.129. URL https://aclanthology.org/2023.acl-short.129.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1314–1331, 2020. doi: 10.1109/SP40000.2020.00095.

Rrubaa Panchendrarajan and Suman Bhoi. Dataset reconstruction attack against language models. In *CEUR Workshop*, 2021.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. Bolt: Privacy-preserving, accurate and efficient inference for transformers. *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 4753–4771, 2024. URL https://api.semanticscholar.org/CorpusID:266178755.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *ArXiv*, abs/1610.05755, 2016. URL https://api.semanticscholar.org/CorpusID:8696462.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *ArXiv*, abs/1802.08908, 2018. URL https://api.semanticscholar.org/CorpusID:3544583.

Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Úlfar Erlingsson. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy. 2019. URL https://api.semanticscholar.org/CorpusID:213343140.

Rahil Parikh, Christophe Dupuy, and Rahul Gupta. Canary extraction in natural language understanding models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 552–560, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.61. URL https://aclanthology.org/2022.acl-short.61.

Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *ArXiv*, abs/2309.17410, 2023. URL https://api.semanticscholar.org/CorpusID:263311025.

Andrew J. Paverd and Andrew C. Martin. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. 2014. URL https://api.semanticscholar.org/CorpusID:211141069.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *ArXiv*, abs/2310.07579, 2023. URL https://api.semanticscholar.org/CorpusID:263834631.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers, 2021. URL https://arxiv.org/abs/2102.01454.

Natalia Ponomareva, Hussein Hazimeh, Alexey Kurakin, Zheng Xu, Carson E. Denison, H. B. McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *J. Artif. Intell. Res.*, 77:1113–1201, 2023. URL https://api.semanticscholar.org/CorpusID:257255451.

Aman Priyanshu, Supriti Vijay, Ayush Kumar, Rakshit Naidu, and Fatemehsadat Mireshghallah. Are chatbots ready for privacy-sensitive applications? an investigation into input regurgitation and prompt-induced sanitization. *arXiv preprint arXiv:2305.15008*, 2023.

Dmitry Prokhorenkov and Yang Cao. Towards benchmarking privacy risk for differential privacy: A survey. pp. 322–327, 11 2023. doi: 10.1145/3600100.3625373.

Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. Natural language understanding with privacy-preserving bert. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021. URL https://api.semanticscholar.org/CorpusID:237260201.

Michael O. Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptol. ePrint Arch.*, 2005:187, 2005. URL https://api.semanticscholar.org/CorpusID:15222660.

Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H. Brendan McMahan, and Françoise Beaufays. Training production language models without memorizing user data, 2020.

Niranjan K. Ray, Deepak Puthal, and Dhruva Ghai. Federated learning. *IEEE Consumer Electronics Magazine*, 10(6):106–107, 2021. doi: 10.1109/MCE.2021.3094778.

Donghwan Rho, Taeseong Kim, Minje Park, Jung Woo Kim, Hyunsik Chae, Jung Hee Cheon, and Ernest K. Ryu. Encryption-friendly llm architecture. 2024. URL https://api.semanticscholar.org/CorpusID:273098659.

Maria Rigaki and Sebastián García. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56:1 – 34, 2020. URL https://api.semanticscholar.org/CorpusID:220525609.

Aleksandr Romanov and Anastasia Fedotova. Natural text anonymization using universal transformer with a self-attention. 2020. URL https://api.semanticscholar.org/CorpusID:211220030.

Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998. URL https://api.semanticscholar.org/CorpusID:2181340.

Timo Schick and Hinrich Schütze. Few-shot text generation with natural language instructions. In *Conference on Empirical Methods in Natural Language Processing*, 2020. URL https://api.semanticscholar.org/CorpusID:238260199.

Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy, 2024.

Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, nov 1979. ISSN 0001-0782. doi: 10.1145/359168.359176. URL https://doi.org/10.1145/359168.359176.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. doi: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x.

Hanyin Shao, Jie Huang, Shen Zheng, and Kevin Chen-Chuan Chang. Quantifying association capabilities of large language models and its implications on privacy leakage. *arXiv preprint arXiv:2305.12707*, 2023.

Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview. *arXiv preprint arXiv:2004.08900*, 2020.

Xinyue Shen, Zeyuan Johnson Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *ArXiv*, abs/2308.03825, 2023. URL https://api.semanticscholar.org/CorpusID:260704242.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2023.

R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2016. URL https://api.semanticscholar.org/CorpusID:10488675.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pp. 1310–1321, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450338325. doi: 10.1145/2810103.2813687. URL https://doi.org/10.1145/2810103.2813687.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. Knowledge unlearning for llms: Tasks, methods, and challenges, 2023.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, 2013. URL https://api.semanticscholar.org/CorpusID:990233.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release strategies and the social impacts of language models, 2019.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

Congzheng Song and Ananth Raghunathan. Information leakage in embedding models, 2020.

Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 196–206, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330885. URL https://doi.org/10.1145/3292500.3330885.

Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248, 2013. URL https://api.semanticscholar.org/CorpusID:8085841.

Alessandro Sordoni, Xingdi Yuan, Marc-Alexandre Côté, Matheus Pereira, Adam Trischler, Ziang Xiao, Arian Hosseini, Friederike Niedtner, and Nicolas Le Roux. Joint prompt optimization of stacked llms using variational inference, 2023.

Samuel Sousa and Roman Kern. How to keep text private? a systematic review of deep learning methods for privacy-preserving natural language processing, 2022.

Nishant Subramani, Sasha Luccioni, Jesse Dodge, and Margaret Mitchell. Detecting personal information in training corpora: an analysis. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pp. 208–220, 2023.

Xianghui Sun, Yunjie Ji, Baochang Ma, and Xiangang Li. A comparative study between full-parameter and lora-based fine-tuning on chinese instruction data for instruction following large language model. *ArXiv*, abs/2304.08109, 2023. URL https://api.semanticscholar.org/CorpusID:258179474.

Kunal Suri, Prakhar Mishra, Saumajit Saha, and Atul Singh. Suryakiran at mediqa-sum 2023: Leveraging lora for clinical dialogue summarization, 2023.

Latanya Sweeney. Guaranteeing anonymity when sharing medical data, the datafly system. *Proceedings : a conference of the American Medical Informatics Association. AMIA Fall Symposium*, pp. 51–5, 1997. URL https://api.semanticscholar.org/CorpusID:28334733.

Xinyu Tang, Richard Shin, Huseyin A. Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. Privacy-preserving in-context learning with differentially private few-shot generation, 2024.

Harry Chandra Tanuwidjaja, Rakyong Choi, and Kwangjo Kim. A survey on deep learning techniques for privacy-preserving. In *International Conference on Machine Learning for Cyber Security*, 2019. URL https://api.semanticscholar.org/CorpusID:202640293.

Harry Chandra Tanuwidjaja, Rakyong Choi, Seunggeun Baek, and Kwangjo Kim. Privacy-preserving deep learning on machine learning as a service—a comprehensive survey. *IEEE Access*, 8:167425–167447, 2020. URL https://api.semanticscholar.org/CorpusID:221725546.

Muhammad Imran Tariq, Nisar Ahmed Memon, Shakeel Ahmed, Shahzadi Tayyaba, Muhammad Tahir Mushtaq, Natash Ali Mian, Muhammad Imran, and Muhammad Waseem Ashraf. A review of deep learning security and privacy defensive techniques. *Mob. Inf. Syst.*, 2020:6535834:1–6535834:18, 2020. URL https://api.semanticscholar.org/CorpusID:215415140.

Apple Differential Privacy Team. Learning with privacy at scale differential. 2017. URL https://api.semanticscholar.org/CorpusID:43986173.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar,

Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen

Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama,

Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

Om Thakkar, Swaroop Indra Ramaswamy, Rajiv Mathews, and Franccoise Beaufays. Understanding unintended memorization in federated learning. *ArXiv*, abs/2006.07490, 2020. URL https://api.semanticscholar.org/CorpusID:219686873.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. Lamda: Language models for dialog applications, 2022.

Zhiliang Tian, Ying Zhao, Ziyue Huang, Yu-Xiang Wang, Nevin Lianwen Zhang, and He He. Seqpate: Differentially private text generation via knowledge distillation. In *Neural Information Processing Systems*, 2022. URL https://api.semanticscholar.org/CorpusID:258509296.

Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. Inferdpt: Privacy-preserving inference for black-box large language model, 2024.

P. Torremans. *Copyright Law: A Handbook of Contemporary Research*. Research handbooks in intellectual property. Edward Elgar, 2007. ISBN 9781848447097. URL https://books.google.it/books?id=X7NEPgAACAAJ.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). *ArXiv*, abs/2011.11660, 2020. URL https://api.semanticscholar.org/CorpusID:227152280.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Considerations for differentially private learning with large-scale public pretraining. *ArXiv*, abs/2212.06470, 2022. URL https://api.semanticscholar.org/CorpusID:254591584.

Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Position: Considerations for differentially private learning with large-scale public pretraining, 2024. URL https://arxiv.org/abs/2212.06470.

Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and YiKe Guo. Privacy preservation in federated learning: An insightful survey from the gdpr perspective. *Computers & Security*, 110:102402, 2021. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2021.102402. URL https://www.sciencedirect.com/science/article/pii/S0167404821002261.

United Nations. *Universal Declaration of Human Rights*. December 1948.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

Arpita Vats, Zhe Liu, Peng Su, Debjyoti Paul, Yingyi Ma, Yutong Pang, Zeeshan Ahmed, and Ozlem Kalinli. Recovering from privacy-preserving masking with large language models, 2023.

Michael Veale, Reuben Binns, and Lilian Edwards. Algorithms that remember: model inversion attacks and data protection law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2133):20180083, 2018.

Davide Venditti, Elena Sofia Ruzzetti, Giancarlo A. Xompero, Cristina Giannone, Andrea Favalli, Raniero Romagnoli, and Fabio Massimo Zanzotto. Enhancing data privacy in large language models through private association editing, 2024. URL https://arxiv.org/abs/2406.18221.

Boxin Wang, Yibo Zhang, Yuan Cao, Bo Li, H. B. McMahan, Sewoong Oh, Zheng Xu, and Manzil Zaheer. Can public large language models help private cross-device federated learning? *ArXiv*, abs/2305.12132, 2023. URL https://api.semanticscholar.org/CorpusID:258833462.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models, 2024a.

Hua Wang, Sheng Gao, Huanyu Zhang, Milan Shen, and Weijie J. Su. Analytical composition of differential privacy via the edgeworth accountant, 2022.

Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models, 2024b. URL https://arxiv.org/abs/2402.05813.

Benjamin Weggenmann, Valentin Rublack, Michael Andrejczuk, Justus Mattern, and Florian Kerschbaum. Dp-vae: Human-readable text anonymization for online reviews with differentially private variational autoencoders. *Proceedings of the ACM Web Conference 2022*, 2022. URL https://api.semanticscholar.org/CorpusID:248367480.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Sean Welleck, Zixin Yao, Yu Gai, Jialin Mao, Zheng Zhang, and Kyunghyun Cho. Loss functions for multiset prediction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/fb3f76858cb38e5b7fd113e0bc1c0721-Paper.pdf.

Sam Witteveen and Martin Andrews. Paraphrasing with large language models. In Alexandra Birch, Andrew Finch, Hiroaki Hayashi, Ioannis Konstas, Thang Luong, Graham Neubig, Yusuke Oda, and Katsuhito Sudoh (eds.), *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 215–220, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5623. URL https://aclanthology.org/D19-5623.

Fan Wu, Huseyin A Inan, Arturs Backurs, Varun Chandrasekaran, Janardhan Kulkarni, and Robert Sim. Privately aligning language models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3d0OmYTNui.

Tong Wu, Ashwinee Panda, Jiachen T. Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models, 2023a.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. Depn: Detecting and editing privacy neurons in pretrained language models, 2023b.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1), aug 2023a. ISSN 0360-0300. doi: 10.1145/3603620. URL https://doi.org/10.1145/3603620.

Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. A differentially private text perturbation method using a regularized mahalanobis metric, 2020.

Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. On a utilitarian approach to privacy preserving text generation, 2021.

Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher A. Choquette-Choo, Peter Kairouz, H. Brendan McMahan, Jesse Rosenstock, and Yuanbo Zhang. Federated learning of gboard language models with differential privacy, 2023b.

Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pp. 160–164, 1982. doi: 10.1109/SFCS.1982.38.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8403–8419, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.457. URL https://aclanthology.org/2024.acl-long.457.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *ArXiv*, abs/2312.02003, 2023. URL https://api.semanticscholar.org/CorpusID:265609409.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Hongxu Yin, Arun Mallya, Arash Vahdat, José Manuel Álvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16332–16341, 2021. URL https://api.semanticscholar.org/CorpusID:233241017.

Ying Yin and Ivan Habernal. Privacy-preserving models for legal natural language processing. In Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Cătălina Goanță, and Daniel Preoțiuc-Pietro (eds.), *Proceedings of the Natural Legal Language Processing Workshop 2022*, pp. 172–183, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.nllp-1.14. URL https://aclanthology.org/2022.nllp-1.14.

Yeojoon Youn, Zihao Hu, Juba Ziani, and Jacob Abernethy. Randomized quantization is all you need for differential privacy in federated learning, 2023.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Gosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in pytorch. *ArXiv*, abs/2109.12298, 2021. URL https://api.semanticscholar.org/CorpusID:237940511.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. *ArXiv*, abs/2110.06500, 2021a. URL https://api.semanticscholar.org/CorpusID:238743879.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. *ArXiv*, abs/2106.09352, 2021b. URL https://api.semanticscholar.org/CorpusID:235458199.

Da Yu, Sivakanth Gopi, Janardhan Kulkarni, Zi-Han Lin, Saurabh Naik, Tomasz L. Religa, Jian Yin, and Huishuai Zhang. Selective pre-training for private fine-tuning. *ArXiv*, abs/2305.13865, 2023. URL https://api.semanticscholar.org/CorpusID:258841179.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. Differential privacy for text analytics via natural text sanitization, 2021.

Xiang Yue, Huseyin A. Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Huan Sun, David Levitan, and Robert Sim. Synthetic text generation with differential privacy: A simple and practical recipe. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:253116660.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pp. 363–375, 2020.

Santiago Zanella-Beguelin, Shruti Tople, Andrew Paverd, and Boris Köpf. Grey-box extraction of natural language models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12278–12286. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zanella-beguelin21a.html.

Collin Zhang, John X. Morris, and Vitaly Shmatikov. Extracting prompts by inverting llm outputs, 2024a. URL https://arxiv.org/abs/2405.15012.

Dan Zhang, Ryan McKenna, Ios Kotsogiannis, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Ektelo: A framework for defining differentially-private computations. *Proceedings of the 2018 International Conference on Management of Data*, 2018. URL https://api.semanticscholar.org/CorpusID:5033862.

Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions, 2024b. URL https://arxiv.org/abs/2307.03941.

Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. Membership inference attacks cannot prove that a model was trained on your data, 2024c. URL https://arxiv.org/abs/2409.19798.

Rui Zhang, Song Guo, Junxiao Wang, Xin Xie, and Dacheng Tao. A survey on gradient inversion: Attacks, defenses and future directions. *ArXiv*, abs/2206.07284, 2022a. URL https://api.semanticscholar.org/CorpusID:249674534.

Rui Zhang, Song Guo, Junxiao Wang, Xin Xie, and Dacheng Tao. A survey on gradient inversion: Attacks, defenses and future directions, 2022b.

Ruisi Zhang, Seira Hidano, and Farinaz Koushanfar. Text revealer: Private text reconstruction via model inversion attacks against transformers. *arXiv preprint arXiv:2209.10505*, 2022c.

Yi Zhang, Yunfan Lu, and Fengxia Liu. A systematic survey for differential privacy techniques in federated learning. *Journal of Information Security*, 14:111–135, 01 2023a. doi: 10.4236/jis.2023.142008.

Zhexin Zhang, Jiaxin Wen, and Minlie Huang. ETHICIST: Targeted training data extraction through loss smoothed soft prompting and calibrated confidence estimation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12674–12687, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.709. URL https://aclanthology.org/2023.acl-long.709.

Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu an Tan. Secure multi-party computation: Theory, practice and applications. *Information Sciences*, 476:357–372, 2019a. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2018.10.024. URL https://www.sciencedirect.com/science/article/pii/S0020025518308338.

Jingwen Zhao, Yunfang Chen, and Wei Zhang. Differential privacy preservation in deep learning: Challenges, opportunities and solutions. *IEEE Access*, 7:48901–48911, 2019b. URL https://api.semanticscholar.org/CorpusID:131776729.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, pp. 159–168, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385268. doi: 10.1145/3462757.3466088. URL https://doi.org/10.1145/3462757.3466088.

Mengxin Zheng, Qian Lou, and Lei Jiang. Primer: Fast private transformer inference on encrypted data. *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2023. URL https://api.semanticscholar.org/CorpusID:257757042.

Nina Zhou, Qiucheng Wu, Zewen Wu, Simeone Marino, and Ivo D. Dinov. Datasiftertext: Partially synthetic text generation for sensitive clinical notes. *Journal of Medical Systems*, 46(12):96, Nov 2022. ISSN 1573-689X. doi: 10.1007/s10916-022-01880-6. URL https://doi.org/10.1007/s10916-022-01880-6.

Junyi Zhu and Matthew B. Blaschko. R-gap: Recursive gradient attack on privacy. *ArXiv*, abs/2010.07733, 2020. URL https://api.semanticscholar.org/CorpusID:222378211.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:195316471.

Yuqing Zhu and Yu-Xiang Wang. Adaptive private-k-selection with adaptive k and application to multi-label pate, 2022.

Zheming Zuo, Matthew Watson, David Budgen, Robert Hall, Chris Kennelly, and Noura Al Moubayed. Data anonymization for pervasive health care: Systematic literature mapping study. *JMIR Med Inform*, 9(10): e29871, Oct 2021. ISSN 2291-9694. doi: 10.2196/29871. URL https://doi.org/10.2196/29871.

A Going Further

As shown throughout this work, the topics discussed here have been subject to extensive research in recent years. In our work, we have shown how privacy-preserving techniques like anonymization or Differential Privacy have been used to build private Language Models. For those readers interested to deepen their knowledge on a single subjects, we present here a list of surveys specific for each single topic:

- **Data Leakage**: Jegorova et al. (2022) present a comprehensive survey on privacy leakage at inference time. They cover involuntary leakage and adversarial leakage and also touch on the available defense mechanisms. The main difference with our work is the fact that we are focused on LLMs and that is also reflected on the techniques shown as solutions which are much more recent.
- **Attacks**: There are several surveys investigating attacks. Rigaki & García (2020) investigate the privacy vulnerabilities of machine learning in general, while some other works are more focused on a specific topic. For example, Enthoven & Al-Ars (2020); Zhang et al. (2022b); Liu et al. (2022) focus on attacks against federated learning models or Derner et al. (2023); Li et al. (2023a) focus on security risks of LLMs, including possible attacks and mitigations. While some of these works have a partial overlap with our section §3, we mostly focus our investigation on *privacy attacks* against LLMs and possible solutions.
- **Privacy Preserving Deep Learning**: Literature yields many works on the topic of privacy preserving deep learning. Some of them are generic, without focusing on any specific model. This is the case for Boulemtafes et al. (2020); Tanuwidjaja et al. (2020); Tariq et al. (2020); Liu et al. (2021); Tanuwidjaja et al. (2019); Mirshghallah et al. (2020) which all touch on privacy issues in deep learning and possible solutions. Although these works are relevant, they primarily focus on deep learning in general and do not address the recent advancements brought by LLMs. Recently, Yao et al. (2023) provide a systematic review on security and privacy in language models divided into three sections: good applications, bad applications, and security risks and solutions. This work is complementary to our study since it covers a broader range of risks (not just privacy risks) while allocating less space to solutions. Ultimately, works focused on privacy in NLP, such as Sousa & Kern (2022), concentrate on NLP and anonymization, while Guerra-Manzanares et al. (2023) targets medical applications without necessarily focusing on textual data.
- **Machine Unlearning**: There are several surveys on the topic of Machine Unlearning: Xu et al. (2023a); Nguyen et al. (2022); Shaik et al. (2024) are general, while Si et al. (2023); Liu et al. (2024a) only focus on LLMs and the differences in application.
- **Differential Privacy**: With the growing interest in Differential Privacy in recent years, numerous literature reviews have emerged. They all focus on different aspects of Differential Privacy: Prokhorenkov & Cao (2023) are focused on the privacy risks of Differential Privacy, how to define it and measure it. Several works investigated the intersection between Differential Privacy and deep learning like Baraheem & Yao (2022); Zhao et al. (2019b); Ponomareva et al. (2023); Demelius et al. (2023): Chen (2023) focus on generative models in particular; Klymenko et al. (2022) focus on DP in the context of NLP; Garrido et al. (2022) present a particularly interesting work as it focuses on the industry, the difficulties in the adoption and gives some suggestions to close the gap between industry and academia; Ouadrhiri & Abdelhadi (2022); Zhang et al. (2023a); Odera (2023) focus on the applications in federated learning.
- **Federated Learning**: Regarding federated learning, several surveys are available with different perspectives. Kairouz et al. (2021b) propose a general survey on federated learning, examining the state of the art and open problems. Hasan (2023); Ma et al. (2022); Truong et al. (2021) give an overview on the privacy and security issues, with Truong et al. (2021) also putting it in relation to GDPR. Lastly Brauneck et al. (2023) give an overview of FL in medical research.
- **Large Language Models**: Chang & Bergen (2023); Zhao et al. (2023); Minaee et al. (2024) present comprehensive surveys on large language models (LLMs), covering various aspects such as

their architecture, training techniques, applications, and the challenges they face. These surveys provide valuable insights into the current state of LLM research, highlight recent advancements, and discuss future directions for the development and deployment of these models.

- **Anonymization** For text data anonymization, the most important works are related to the healthcare domain, which is the case for Larbi et al. (2022); Zuo et al. (2021); Meystre et al. (2010).

B Summary Tables

In this section we present summary tables for all the approaches discussed in the text. Table 2 and Table 3 summarize the anonymization techniques respectively presented in §4.1 and §**??**. Table 4 presents some information about the works that perform a fully private training, discussed in §5.1.1, with information about the model, the datasets used both for pre-training and fine-tuning and the optimization method. Similarly in Table 5 there is an overview of the works that perform private fine-tuning, discussed in §5.1.2. Table 6 summarizes the works found in literature for private inference, which are discussed in §5.2. Lastly, Table 7 presents a brief summary of the Unlearning methods presented in §5.4.

Data Anonymization

Method	**Summary**	**Reference**
Modular	Modular system that has pre-processing, NER, CRR and Anonymization. Performances dependent on NER module.	Mamede et al. (2016)
Perturbation	Altering data while preserving statistical significance. Effectively anonymizes data but may compromise the accuracy.	Zuo et al. (2021)
DataSifterText	Sensitive information is masked and then imputed with BERT. Whitelist to mask sensitive terms, blacklist for meaningful terms to retain. Retains high data utility.	Zhou et al. (2022)
Transformer	Text analyzed to extract identifying features, which are then smoothed. Universal Transformer is used to generate text incorporating the features. Integrates modern model with classical approaches. Specific to Russian.	Romanov & Fedotova (2020)
RLTA	Attention-based text representation learner extract latent embeddings and a deep RL-based privacy preserver manipulates them. Effectively balances privacy and utility, but is dependent on precise parameter tuning and has data-specific limitations.	Mosallanezhad et al. (2019)
LLMs	Use an LLM to impute previously masked tokens. They find NEs with NER and then mask them. Performances are significantly better when compared to a model trained on the masked data.	Vats et al. (2023)
HaS	Two local LLMs, the first one anonymizes the prompt before sending it to a black-box LLM, the second one de-anonymizes the output from the black-box LLM. They show a good privacy-utility tradeoff.	Chen et al. (2023b)

Table 2: Classical methods for text anonymization

Data Anonymization with DP

Method	Summary	Reference
ADePT	Clipping and noising the embeddings before decoding to build a DP latent representation. DP proof is problematic.	Krishna et al. (2021)
DP-VAE	Apply isotropic Gaussian noise in the latent space. Good results for authorship anonymization.	Weggenmann et al. (2022)
DP-BART	Clipping and noising the embeddings, new clipping and iterative pruning allows for a smaller encoder. Performances over the baseline.	Igamberdiev & Habernal (2023)
BERT	Purkayastha Mechanism used to take advantage of the angular distance as a metric. Performances close to non-private and increased MIAs robustness.	Du et al. (2023b)
SANTEXT	Words mapped into embeddings, then compute distances and use them to sample replacement inversely proportional to the distance	Yue et al. (2021)
CUSTEXT	Replace every token: mapping function determines output space and sampling function chooses replacement tokens. Good privacy/utilty tradeoff.	Chen et al. (2023a)
RANTEXT	Random adjacency lists to increase embedding inversion attacks robustness. Used with black-box LLMs and a de-anonymization module for open-ended text generation, shows very strong performances	Tong et al. (2024)
biLSTM	New noise distribution specifically designed for DP. Epsilon calibration based on geometric properties of the word embedding space. Shows a favorable privacy/utility tradeoff.	Feyisetan et al. (2019b)
TEM	Truncated Exponential Mechanism to replace standard Exponential Mechanism, which is not suitable for metric DP as it does not consider the specific structure of the embedding space. Substantial improvements over baseline.	Carvalho et al. (2021)
RMM	Regularized Mhalanobis Metric used for text perturbation to add elliptical noise to ensure that words in sparse regions have a sufficient likelihood of replacement without sacrificing overall utility.	Xu et al. (2020)
Syntactic Approach	Include grammatical categories into privatization to preserve grammatical properties. Improves performances but slightly decreases privacy.	Arnold et al. (2023)
Paraphrase	Fine-tuned GPT2, to introduce noise, temperature is sampled based on the exponential mechanism. Superior performance with respect to word-level DP.	Mattern et al. (2022b)

Table 3: Methods for text anonymization with DP

Models pre-trained with DP

Model	Method	Datasets	Reference
BERT	DP-SGD	BERT dataset+MIMIC-III pre-training i2b2-2010 fine-tuning	Hoory et al. (2021)
BERT	DP-SGD SGD	BookCorpus+Wikipedia+Legal pre-training Overruling dataset/CaseHOLD fine-tuning	Yin & Habernal (2022)
Feedforward NN	DP-SGD	Brown corpus pre-training Reddit comments dataset fine-tuning	Kerrigan et al. (2020)
GPT2	DP-SGD	Selected 15% of OpenWebText pre-trainig Enron email dataset fine-tuning	Tramèr & Boneh (2020)

Table 4: Models pre-trained with DP

Models fine-tuned with DP

Model	Method	Dataset	PEFT	Reference
GPT2	DP-SGD	Open Yelp Dataset	✗	Yue et al. (2022)
RoBERTa	DP-ADAM	MNLI,QQP,QNLI,SST-2	✗	Li et al. (2021)
GPT2	DP-ADAM	E2E dataset	✓	Li et al. (2021)
DialoGPT	DP-ADAM	Persona-Chat dataset	✗	Li et al. (2021)
GPT2	DP-SGD	IMDb, Amazon reviews dataset	✗	Mattern et al. (2022a)
RoBERTA	DP-SGD	MNLI, SST-2, QQP, QNLI	✓	Yu et al. (2021a)
GPT2	DP-SGD	E2E, DART	✓	Yu et al. (2021a)
BERT	ADAM	SST, QQP, TP-UK	✓[a]	Li et al. (2023d)
T5	ADAM	SST, QQP, TP-UK	✓[b]	Li et al. (2023a)
GPT2	DP-SGD, DP-PPO	Reddit TL;DR, IMDb	✗	Wu et al. (2024)

[a]The authors used prompt-tuning and prefix-tuning after obfuscating the embeddings
[b]The authors used prompt-tuning and prefix-tuning after obfuscating the embeddings

Table 5: Models fine-tuned with DP

DP-Inference methods

Model	Method	Dataset	Reference
GPT-3	Report-Noisy-Max+GM ESA KSA	SST-2, Amazon, AGNews, TREC, PFL-DocVQA, SAMSum	Wu et al. (2023a)
GPT-3	GM, EM	AGNews, DBPedia, TREC MIT-G, MIT-D	Tang et al. (2024)
GPT-3	PromptPATE	SST-2, AGNews, TREC, DBPedia	Duan et al. (2024a)
Llama-2, GPT-3, Vicuna	DP-OPT	SST-2, TREC, Mpqa, Disaster	Hong et al. (2023)
T5, BERT, GPT-2	Split-and-Denoise	TweetEval Offensive, Hate Speech 18, Healt Fact, Daily Dialogue and others	Mai et al. (2023)
GPT-4, Vicuna	InferDPT	CNN/Daily Mail, Wikitext-103-v1, ArXiv Dataset	Tong et al. (2024)

Table 6: DP-Inference methods

Machine Unlearning Methods

Method	Summary	Reference
ROME	Allows to track factual predictions back to single neurons and edit them individually. Proven to be ineffective in some cases.	Meng et al. (2023a)
MEMIT	With a similar approach with respect to ROME, scale the editing on multiple examples. The discussion is focused on factuality.	Meng et al. (2023a)
DEPN	This method assume that private information is encoded in specific neurons: their weights are set to zero to block information flow.	Wu et al. (2023b)
PAE	Model editing to break the association between a user identity and its PII. The edit method is effective both in batch and in sequential editing.	Venditti et al. (2024)
SISA	Divide training data in shards, further divided in slices and then train instances of the model on each shard by incrementally using slices and saving parameters at every step. Requires limited re-training.	Bourtoule et al. (2019)
KU for privacy	Negate the loss function used in training to maximize the loss on the target sequence (to unlearn)	Jang et al. (2022)
Approximate UL in LLMs	Fine-tune a model on target sequences, then use it to lower the likelihood sequences predicted by this model.	Eldan & Russinovich (2023)
UL layers	Build Unlearning layers and train them with a selective student-teacher objective based on KL-divergence to maximize divergence from teacher model	Chen & Yang (2023)
DeMemorization	Take samples from pre-training dataset, divide in prefix and suffix, use a negative similarity metric to evaluate how different is the generated suffix from the real one. This is used as a reward signal to encourage the LLM to develop a paraphrasing policy	Kassem et al. (2023)
Knowledge Sanitization	Use LoRA to instruct the model with a sanitization phrase when a potentially harmful prompt is given	Ishibashi & Shimodaira (2023)
ICUL	Build a context where the datapoint you want to unlearn is included but with a flipped label	Pawelczyk et al. (2023)

Table 7: Machine Unlearning Methods

C Discussion on Federated Learning

The central model of Differential Privacy (CDP) enforces privacy at the participant level rather than the record level. Record level Differential Privacy corresponds to the classical definition (see §2.2 for more details) and states that the outcome of a differentially private computation should be the same (with probability ϵ) regardless of whether a particular record is or is not in the dataset. As Naseri et al. (2020) explain, participant-level DP ensures (with probability ϵ) that the aggregation function produces the same result regardless of whether a particular client participates in the training process.

Although this method is more secure than sending plain data, it requires a high level of trust in the central server and in the communication channel. Clients send unperturbed gradients, which can lead to training data reconstruction via gradient inversion (Zhang et al., 2022a), and even averaged gradients before perturbation can be exploited for reconstruction, as shown by Geiping et al. (2020), see §3.3 for more information. This means that an honest-but-curious (or semi-honest (Paverd & Martin, 2014)) server could potentially access the training data. Given these risks, Central Differential Privacy is not widely used. However, Naseri et al. (2020) demonstrates that CDP can reduce federated learning's vulnerability to backdoor and white-box Membership Inference Attacks (§3.2). For instance, on the CIFAR-10 dataset, they achieved a 23% reduction in the accuracy of MIAs without a significant loss in utility.

McMahan et al. (2018) introduce DP-FedAvg, a variation of the FedAVG algorithm by McMahan et al. (2016), incorporating the moments accountant (Abadi et al., 2016) to calculate the privacy budget used during training. Typically, FedAVG performs part of the training locally on a subset of clients selected based on specific criteria. The clients then send their updated gradients to the server, which averages them, updates the global model, and sends the updated model back to the clients. In DP-FedAvg, gradients are clipped locally but not obfuscated. The central server adds noise after receiving and aggregating the gradients to prevent inversion, aligning the algorithm with central DP since local clipping alone does not meet the Local DP criteria. This approach was tested by training an LSTM for next token prediction on a large dataset of Reddit posts described by Al-Rfou et al. (2016), using various noise levels and clipping strategies. The model's performance was evaluated using AccuracyTop1, the probability that the predicted word is correct. The study demonstrated that, with sufficient data, it is possible to train a model with privacy guarantees without significantly sacrificing performance, provided the noise added is kept within reasonable limits.

Ramaswamy et al. (2020) apply DP-FedAVG to develop a private Next Word Prediction (NWP) model using an LSTM in a federated setting. Their goal is to train production-quality NWP models with DP-FedAvg in a real-world environment, specifically on a diverse fleet of mobile phones used for Gboard. Unlike DP-FedAVG's typical Poisson sampling, where each user is selected independently with a fixed probability for each round, this work employs fixed-size federated rounds. In this method, a fixed number of users is randomly sampled to participate in each round, ensuring a consistent number of participants and facilitating better management of computational and communication resources in a production environment like Gboard. The model's performance is evaluated on a user-generated dataset from Gboard, using the same private dataset for training, and is measured in terms of Top-1 Recall and Top-3 Recall. The results surpass the non-private baseline N-gram FST (Finite State Transducer). Privacy attacks based on the work of Carlini et al. (2019) are conducted to test the model's resistance to memorizing canaries. The findings show that the model does not memorize canaries unless they are repeated approximately 200 times or shared among roughly 16 users. The authors emphasize that the Differential Privacy approach relies on several assumptions—mostly sound but challenging to verify—highlighting the importance of empirical measurements like canary tests to validate privacy guarantees.

Currently, large language models (LLMs) represent the state of the art due to their superior power, though they are much larger than the small LSTMs seen in previous studies. Wang et al. (2023) focus on using LLMs to enhance the privacy/utility tradeoff of local LMs in differentially private federated learning. Similar to Yu et al. (2023), which samples pre-training data based on its similarity to fine-tuning data, they propose a distribution matching algorithm to sample public data close to the private data distribution, significantly improving sample efficiency. They utilize DP-FTRL (Kairouz et al., 2021a), an online optimization algorithm combining FTL (follow the leader) and regularization, adding noise during local updates. The local LM is either a single-layer LSTM (Hochreiter & Schmidhuber, 1997) or Transformer (Vaswani et al., 2023) due to computational limits on edge devices, while the LLM used is LaMDA (Thoppilan et al., 2022). Their tests yielded several notable results. First, using a public tokenizer with a public vocabulary instead of a private one helps prevent privacy leakage. Second, pre-training local models on a public dataset (C4 dataset) for over a week of single GPU time improved next token prediction accuracy (tested on the private StackOverflow dev set). They also built a distillation corpus by saving the top-k logits with k nonzero entries from the teacher LLM's next token predictions as a silver label dataset. Pre-training on this distillation corpus showed accuracy comparable to pre-training with the original dataset, even with just 1% distillation

coverage. Furthermore, their new distribution matching technique allows using only 0.08% of the pre-training dataset to achieve performance equal to 1% distillation coverage, greatly enhancing sample efficiency.

Xu et al. (2023b) present and analyze twenty Gboard LMs trained for Next Word Prediction using DP-FTRL, asserting that all future Gboard models will have DP guarantees. In their DP-FTRL implementation, model updates are clipped and noise is added during global model updates, ensuring Central Differential Privacy. They also use adaptive clipping (Andrew et al., 2022), which adjusts the clip norm each round by privately estimating the model's norm at a targeted quantile. However, adaptive clipping showed no significant improvement and was less robust in experiments with large report goals, sometimes causing catastrophic failure or no progress in the first 1000 rounds. Despite this, adaptive clipping can reduce hyperparameter tuning when the privacy budget permits. Public pre-training on C4, as explained by Wang et al. (2023), reduced the rounds needed to reach target utility by 1000 under the same noise multiplier, enhancing privacy guarantees. They also combined DP-FTRL with Secure Aggregation (Bonawitz et al., 2017), a cryptographic protocol ensuring the central server only accesses aggregated updates, not individual client updates. While SecAgg aids DP, it introduces significant computational slowdown and requires different system configurations for large report goals. The techniques for training language models with differential privacy in Gboard have produced models with stronger privacy guarantees than some established standards, such as those used by the US Census Bureau. In Europe, these differentially private neural network models outperform older N-gram models in utility. For English in the US and Portuguese in Brazil, the new models are on par with or slightly better than their non-private counterparts.

Basu et al. (2021) provide another example of differentially private training of LMs within a central model in a federated setting. They trained BERT, RoBERTa, DistillBERT, and ALBERT on tweets labelled to detect depression (Depression dataset) and sexual harassment (Sexual harassment dataset). Although the authors do not detail the training procedure, their results show that all models were trained with various epsilon values. Interestingly, the smaller models performed more favourably under these conditions.

C.1 Federated Learning with Local DP

Initially formalized by Kasiviswanathan et al. (2008), local Differential Privacy (LDP) involves clipping and adding noise locally at the clients' level before sending gradients (see Figure 5.3). This method removes the need for a trusted central server and ensures safety even if an eavesdropper intercepts the gradients as they are noisy. It makes the process safer, but it comes with its own challenges. Though primarily used in networks handling image data, LDP approaches are worth mentioning for LLMs, as there is no theoretical barrier to their application. The main obstacle is the computational burden coming from the high dimensionality of text data, which may be mitigated in the future.

Shokri & Shmatikov (2015) introduce DSSGD (Distributed Selective Stochastic Gradient Descent) for private distributed training of MLP and CNN models on MNIST and SVHN datasets for digit classification. While not directly related to language modeling, the algorithm is broadly applicable to various network types. DSSGD builds on SSGD, a local Selective SGD that selectively applies gradients based on certain criteria to reduce the amount of training data information used for updates, thereby enhancing privacy, though not guaranteeing it. In DSSGD, each client trains a local model using SSGD's selection mechanism, and then shares selected parameters with a central server. The server aggregates these parameters to update the global model and redistributes it to clients for further local updates. This approach, however, has potential privacy risks from gradient selection and sharing. To address these, the authors use the Sparse Vector Technique (Dwork et al., 2006a; Hardt & Rothblum, 2010) to randomly select gradients exceeding a threshold and add noise using the Laplacian mechanism. Their experiments show that models trained with DSSGD achieve accuracy close to non-private methods. Specifically, on the MNIST dataset, the accuracy is almost indistinguishable from non-private training. Additionally, even with only 1% of parameters shared, the collaborative model significantly outperforms single local models.

Youn et al. (2023) introduce RQM (Randomized Quantization Mechanism), a novel approach to enforce DP in federated settings with improved communication efficiency, used alongside Distributed SGD on a CNN for image classification on EMNIST. RQM enhances DP-SGD by eliminating the need for explicit noise addition, instead performing RQM after gradient clipping. In RQM, gradient updates are encoded in quantization

levels, which are then randomly subsampled. A randomized rounding procedure maps continuous gradient values to the nearest subsampled quantization level. RQM falls under Renyi Differential Privacy (Mironov, 2017), a relaxation of the original differential privacy definition, placing the method under Local DP since RQM is performed on the clients. Tests show that RQM outperforms the Poisson Binomial Mechanism (Chen et al., 2022b) (a traditional noise injection method) and noise-free clipped SGD, achieving better accuracy with stronger privacy guarantees.

Bebensee (2019) provide a comprehensive overview of Local Differential Privacy (LDP) and its notable implementations, highlighting the increased variance and noise compared to the central model of DP. In LDP, perturbations are performed locally, leading to higher total variance, which directly depends on the number of participants. The lower error bound in the local setting is dependent on n (the number of participants), necessitating a very high number of participants to inject sufficient noise. Additionally, since each client adds its own noise, the total noise is much higher, requiring significantly more data for efficient learning or a higher ϵ.

LDP is widely used in the industry to collect and aggregate telemetry data. While these approaches are not directly relevant to the research presented here, they demonstrate how high-profile companies use DP in daily operations. Google (Erlingsson et al., 2014), Microsoft (Ding et al., 2017), and Apple (Team, 2017) have developed their own LDP algorithms to collect and aggregate usage statistics for applications like Chrome and the Apple keyboard. This also goes to show once again how local DP is widely used, but for low dimensionality data (telemetry data are usually counters), while for high dimensionality data like text there is still no sound implementation.