Analyzing Leakage of Personally Identifiable Information in Language Models





https://nilslukas.github.io

Nils Lukas, Dec 11 2023
Research Presentation @Meta





Nils Lukas



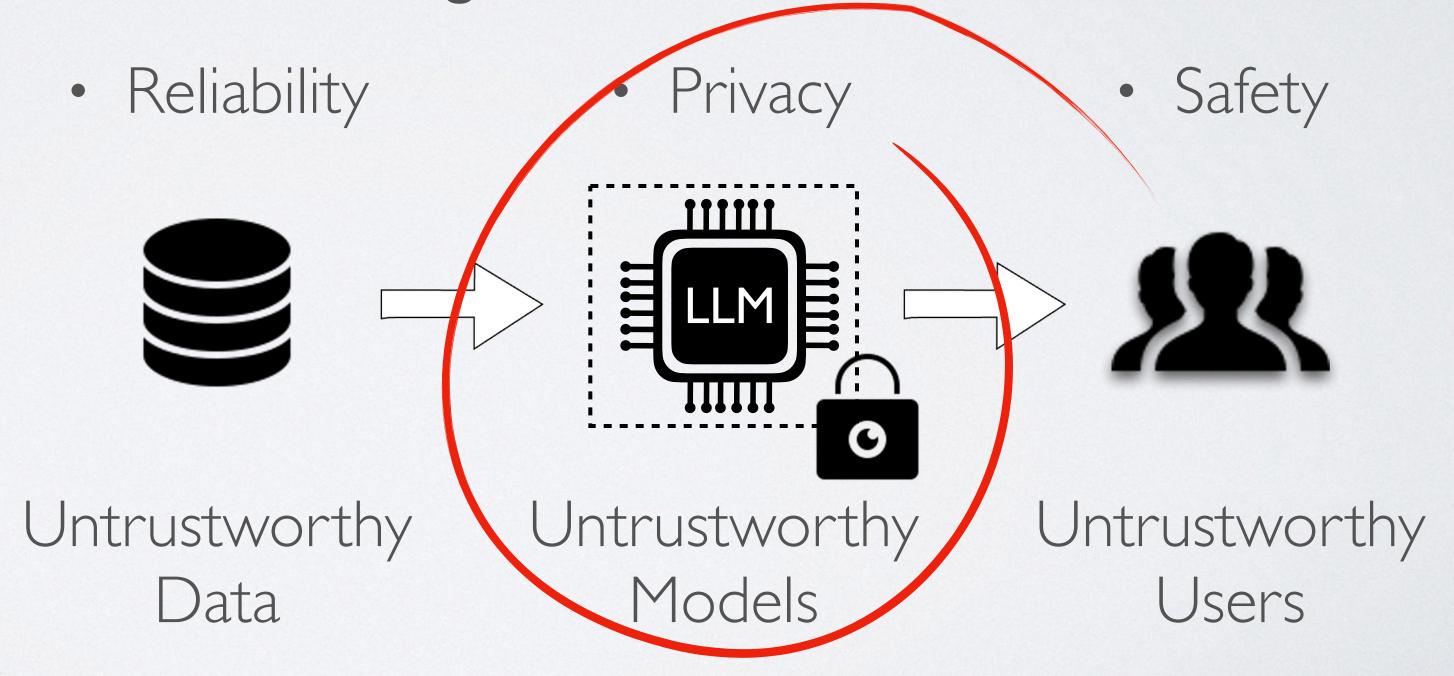
https://nilslukas.github.io

My Areas of Research

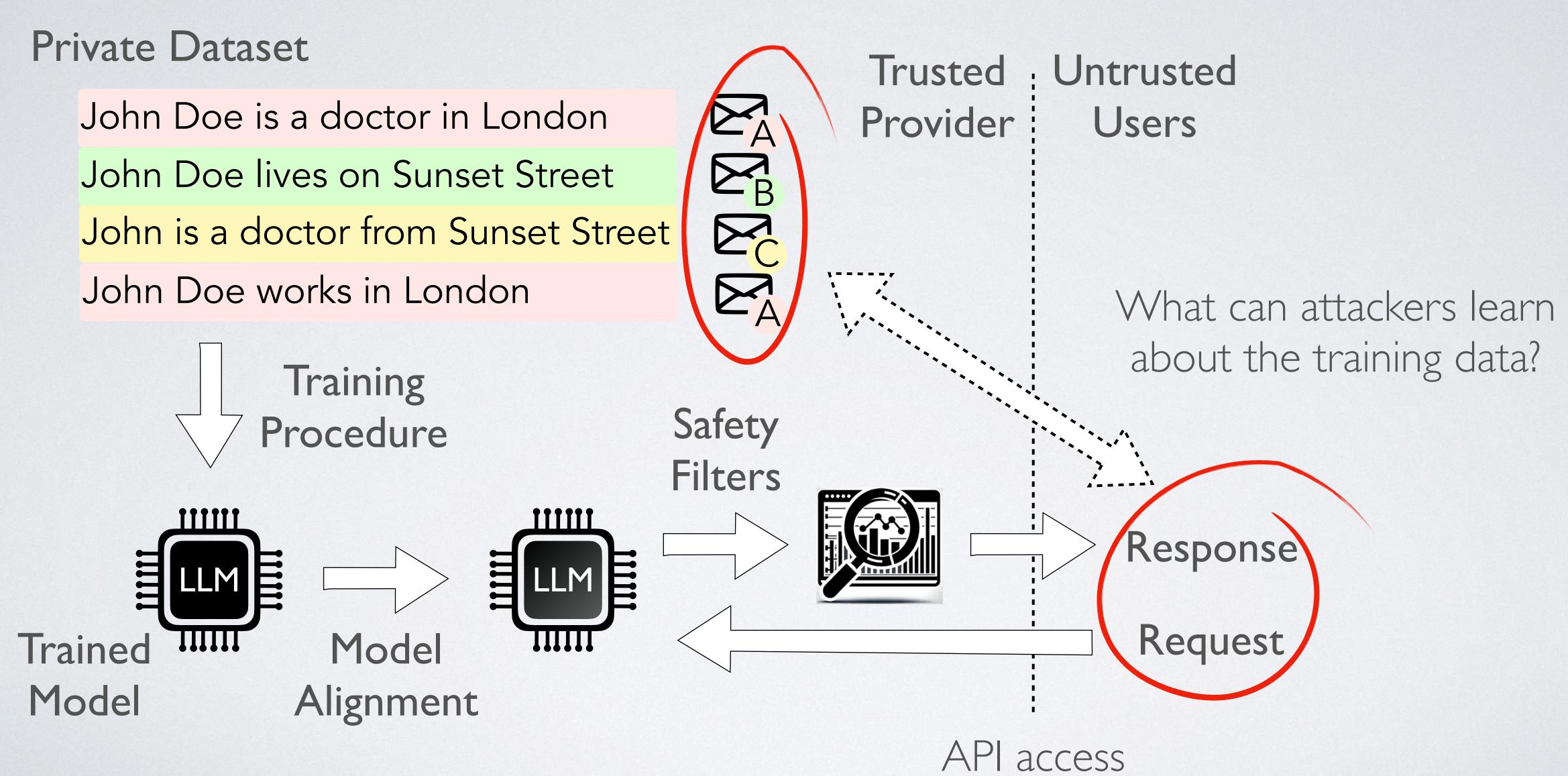
Private Computation

- Private Set Intersection
- Secure Inference

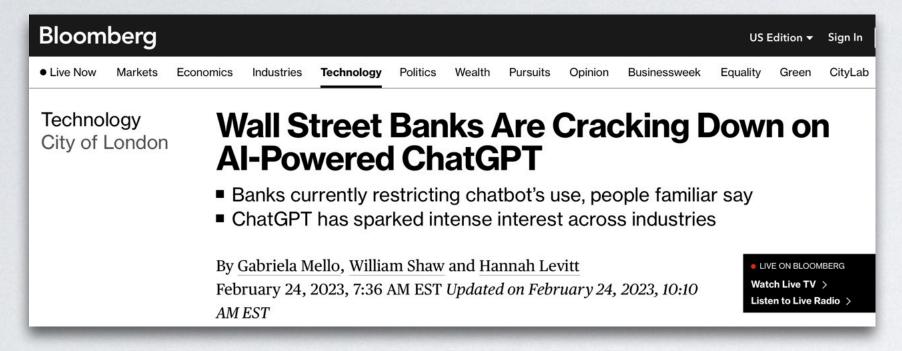
Machine Learning



Data Privacy for Large Language Models



Privacy Concerns



Bloomberg, 2023 [1]

ChatGPT banned in Italy over privacy concerns

(1) 1 April

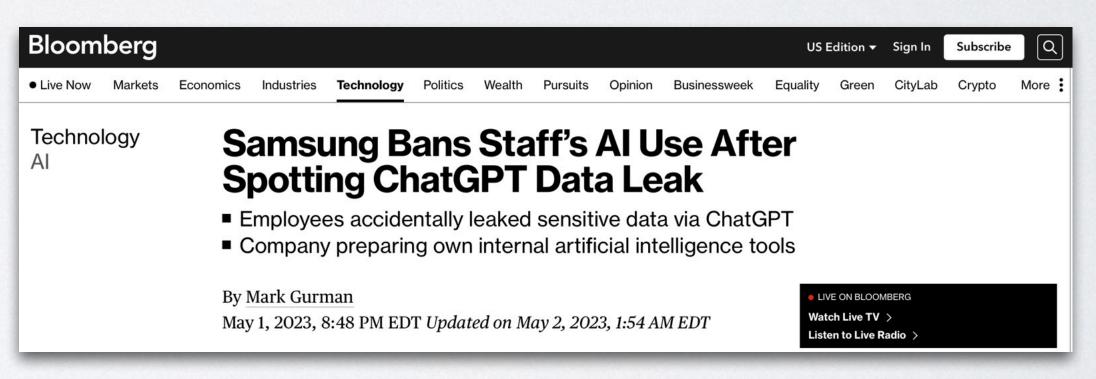
ChatGPT accessible again in Italy

38 April

BBC News, 2023 [3,4]



Business Insider, 2023 [2]



Bloomberg, 2023 [5]

Lack of Privacy in Code Completion

SECURITY

10,000 AWS secret access keys carelessly left in code uploaded to GitHub

By **Shawn Knight** March 25, 2014, 1:00 PM

Techspot, 2014 [9]

GitHub Copilot Al Is Leaking Functional API Keys

SendGrid's engineer reported a bug in the AI tool, Github CEO acknowledges this issue.

By Amit Kulkarni July 29, 2021

Analytics Drift, 2021 [10]

SECRETS DETECTION

Yes, GitHub's Copilot can Leak (Real) Secrets

Researchers successfully extracted valid hard-coded secrets from Copilot and CodeWhisperer, shedding light on a novel security risk associated with the proliferation of secrets.

GitGuardian, 2023 [11]

Terms of Service

6. Will you use my conversations for training?

 Yes. Your conversations may be reviewed by our Al trainers to improve our systems.

ChatGPT by OpenAl [6]

Who has access to my Bard conversations?

We take your privacy seriously and we do not sell your personal information to anyone. To help Bard improve while protecting your privacy, we select a subset of conversations and use automated tools to help remove personally identifiable information. These sample conversations are reviewable by trained reviewers and kept for up to three years, separately from your Google Account.

Please do not include information that can be used to identify you or others in your Bard conversations.

Bard by Google [7]

Privacy Threats

scientific reports

Check for updates

Privacy

GPT-4 has learned from a variety of include publicly available personal about people who have a significant figures. GPT-4 can also synthesize reasoning within a given completic to personal and geographic inforn with a phone number or answering without browsing the internet. For address to a phone number with a as being through that route. By potential to be used to attempt to

OPEN Man vs the machine in the struggle for effective text anonymisation in the age of large language models

Constantinos Patsakis^{1,2,4™} & Nikolaos Lykousas^{2,3,4}

The collection and use of personal data are becoming more common in today's data-driven culture. While there are many advantages to this, including better decision-making and service delivery, it also poses significant ethical issues around confidentiality and privacy. Text anonymisation tries to prune and/or mask identifiable information from a text while keeping the remaining content intact to alleviate privacy concerns. Text anonymisation is especially important in industries like healthcare, law, as well as research, where sensitive and personal information is collected, processed, and exchanged under high legal and ethical standards. Although text anonymisation is widely adopted in practice, it continues to face considerable challenges. The most significant challenge is striking a balance between removing information to protect individuals' privacy while maintaining the text's usability for future purposes. The question is whether these anonymisation methods sufficiently reduce the risk of re-identification, in which an individual can be identified based on the remaining information in the text. In this work, we challenge the effectiveness of these methods and how we perceive identifiers. We assess the efficacy of these methods against the elephant in the room, the use of AI over big data. While most of the research is focused on identifying and removing personal information, there is limited discussion on whether the remaining information is sufficient to deanonymise individuals and, more precisely, who can do it. To this end, we conduct an experiment using GPT over anonymised texts of famous people to determine whether such trained networks can deanonymise them. The latter allows us to revise these methods and introduce a novel methodolog that employs Large Language Models to improve the anonymity of texts.

In today's data-driven society, the collection and use of personal information are becoming increasingly preva lent. While this has numerous benefits, such as improved decision-making and better service provision, it also raises important ethical concerns related to privacy and confidentiality. Indeed, harvesting user data is a common practice of far too many online platforms and services with a significant impact on citizens. This has been ain its reasoning one of the pillars that led to the introduction of the General Data Protection Regulation (GDPR)1 and other relevant legislation around the world as a means to address the privacy issues that emerged. The GDPR mandates using privacy-preserving methods and processes throughout the data management lifecycle, from collection and processing to sharing and publishing. One of these fundamental methods is anonymisation. Given that modern organisations continuously deal with documents, the above has served as a catalyst in the emergence of text

de data. anonymisation as a research topic with many practical applications. The general concept is that given a text, one has to remove or mask identifiable information while preserving the remaining content. Text anonymisation is particularly relevant in healthcare, law, and research, where personal and sensitive information is overwhelming and must be protected to comply with privacy regulations and ethical guidelines.

Although text anonymisation has been widely adopted in practice, it still faces significant challenges. These methods must strike a balance between the need to protect the privacy of individuals and the need to preserve the data utility. Let us consider this with an example where the anonymisation task is to anonymise the sentence "Volodymyr Zelenskyy is the president of Ukraine". Clearly, simply removing the name is not enough. If one is given the sentence "NAME is the president of Ukraine", it is trivial to recover the missing information. Therefore, the anonymised sentence would be " NAME is the president of COUNTRY ". To this end, named entity

¹Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou str, 18534 Piraeus, Greece. ²Management Systems Institute of Athena Research Centre, Marousi, Greece. 3Data Centric Services, Bucharest, Romania. 4These authors contributed equally: Constantinos Patsakis and Nikolaos Lykousas. ™email: kpatsak@unipi.gr

Scientific Reports | (2023) 13:16026

https://doi.org/10.1038/s41598-023-42977-3

have knowledge cities and public multiple steps of that may relate tions associated completion and Jniversity email GPT-4 has the

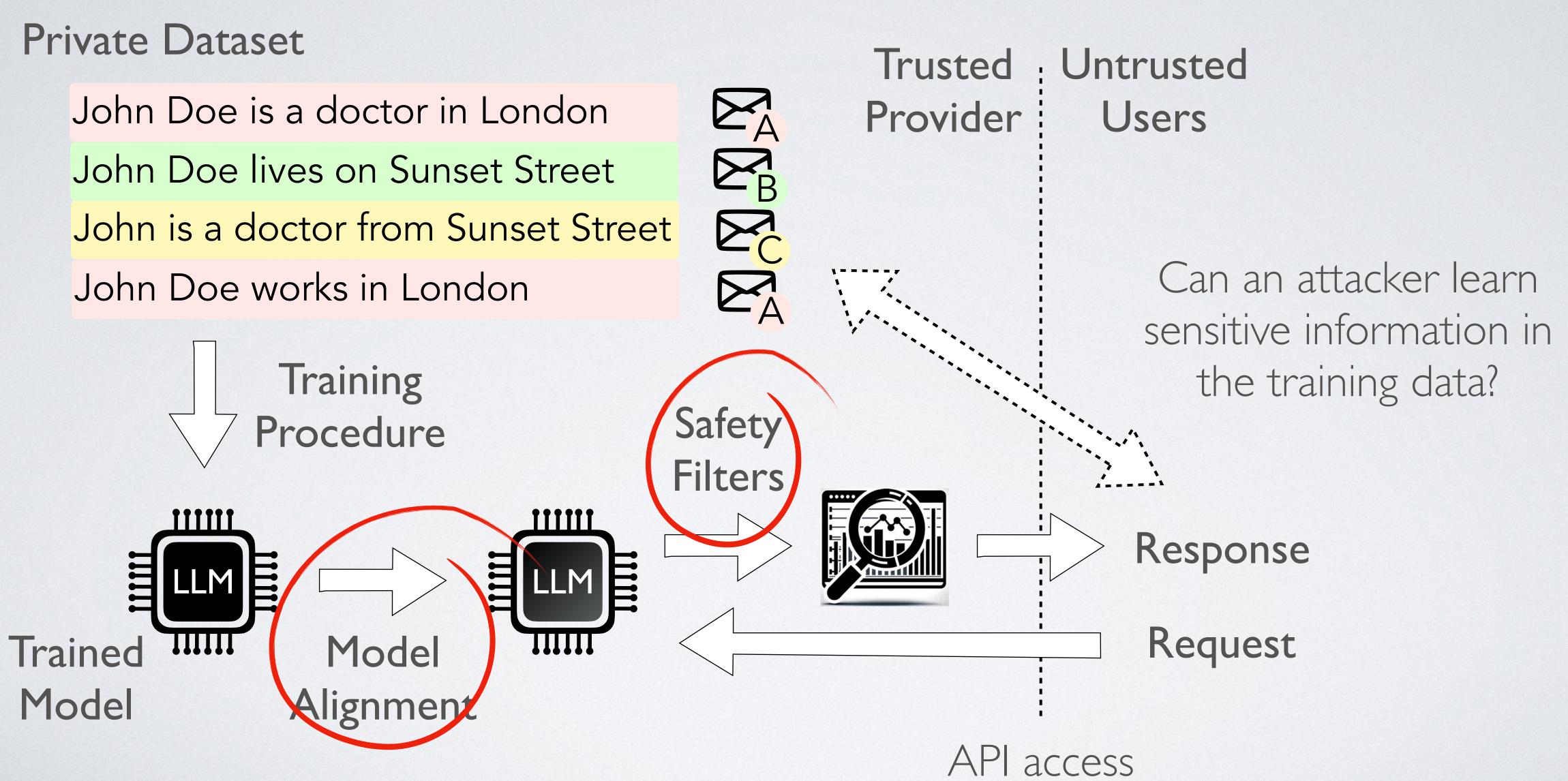
irces, which may

[B] GPT-4 Technical Report, OpenAl., Preprint, March 2023

Augmented Data Attack [C]

[C] Man vs the Machine in the Struggle for Effective Text Anonymization in the Age of Large Language Models, Patsakis et al., Scientific Reports

Data Privacy for Large Language Models



Privacy Attacks by Evading Model Alignment

Multi-step Jailbreaking Privacy Attacks on ChatGPT

Haoran Li*1, Dadi Guo*2, Wei Fan1, Mingshi Xu1, Jie Huang³, Fanpu Meng⁴, Yangqiu Song¹

¹Dept. of CSE, Hong Kong University of Science and Technology ²Center for Data Science, AAIS, Peking University ³Dept. of Computer Science, University of Illinois at Urbana-Champaign ⁴The Law School, University of Notre Dame

{hlibt, wfanag, mxuax}@connect.ust.hk,guodadi@stu.pku.edu.cn jeffhj@illinois.edu, fmeng2@nd.edu, yqsong@cse.ust.hk

With the rapid progress of large language models (LLMs), many downstream NLP tasks can be well solved given appropriate prompts. Though model developers and researchers work hard on dialog safety to avoid generating harmful content from LLMs, it is still challenging to steer AI-generated content (AIGC) for the human good. As powerful LLMs are devouring existing text data from various domains (e.g. GPT-3 is trained on 45TB texts), it is natural to doubt whether the private information is included in the training data and what privacy threats can these LLMs and their downstream applications bring. In this paper, we study the privacy threats from OpenAI's ChatGPT and the New Bing enhanced by ChatGPT and show that application-integrated LLMs may cause new privacy threats. To this end, we conduct extensive experiments to support our claims and discuss LLMs' privacy implications.

19

05

The rapid evolution of large language models ern natural language processing. LLMs' domiparadigms to a unified text generation task and coninstructions/prompts, LLMs even can be zero-shot or few-shot learners to solve specified tasks (Chen Wei et al., 2022b: Sanh et al., 2022).

Massive LLMs' textual training data are primar- LLMs we use. ily collected from the Internet and researchers pay less attention to the data quality and confidentiality of the web-sourced data (Piktus et al., 2023).

Haoran Li and Dadi Guo contribute equally.

Such mass collection of personal data incurs debates and worries. For example, under the EU's General Data Protection Regulation (GDPR), training a commercial model on extensive personal data without notice or consent from data subjects lacks a legal basis. Consequently, Italy once temporarily banned ChatGPT due to privacy considerations1.

Unfortunately, the privacy analysis of language models is still less explored and remains an active area. Prior works (Lukas et al., 2023; Pan et al., 2020; Mireshghallah et al., 2022; Huang et al., 2022; Carlini et al., 2021) studied the privacy leakage issues of language models (LMs) and claimed that memorizing training data leads to private data leakage. However, these works mainly investigated variants of GPT-2 models (Radford et al., 2019) trained simply by language modeling objective, which aimed to predict the next word given the current context. Despite the efforts made by these pioneering works, there is still a huge gap between the latest LLMs and GPT-2. First, LLMs' model sizes and dataset scales are much larger than GPT-2. Sec-(LLMs) makes them a game changer for mod- ond, LLMs implement more sophisticated training objectives, which include instruction tuning (Wei nating generation ability changes previous tasks' et al., 2022a) and Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017). sistently improves LLMs' performance on these Third, most LLMs only provide application protasks (Raffel et al., 2020; Chung et al., 2022; Brown gramming interfaces (APIs) and we cannot inspect et al., 2020b; OpenAI, 2023; Ouyang et al., 2022; the model weights and training corpora. Lastly, it Chan et al., 2023). Moreover, given appropriate is trending to integrate various applications into LLMs to empower LLMs' knowledge grounding ability to solve math problems (ChatGPT + Wolet al., 2021; Zhou et al., 2023; Kojima et al., 2022; fram Alpha), read formatted files (ChatPDF), and respond to queries with the search engine (the New Notably, LLMs' training data also scale up in Bing). As a result, it remains unknown to what accordance with models' sizes and performance. extent privacy leakage occurs on these present-day

To fill the mentioned gap, in this work, we con-

technology-65139406. Currently, ChatGPT is no longer banned in Italy.

Jailbroken: How Does LLM Safety Training Fail?

Content Warning: This paper contains examples of harmful language.

Alexander Wei UC Berkeley awei@berkeley.edu

202

5

Xiv:2307.02483v1

Nika Haghtalab UC Berkeley nika@berkeley.edu

Jacob Steinhardt* UC Berkeley jsteinhardt@berkeley.edu

Abstract

Large language models trained for safety and harmlessness remain susceptible to adversarial misuse, as evidenced by the prevalence of "jailbreak" attacks on early releases of ChatGPT that elicit undesired behavior. Going beyond recognition of the issue, we investigate why such attacks succeed and how they can be created. We hypothesize two failure modes of safety training: competing objectives and mismatched generalization. Competing objectives arise when a model's capabilities and safety goals conflict, while mismatched generalization occurs when safety training fails to generalize to a domain for which capabilities exist. We use these failure modes to guide jailbreak design and then evaluate state-of-the-art models, including OpenAI's GPT-4 and Anthropic's Claude v1.3, against both existing and newly designed attacks. We find that vulnerabilities persist despite the extensive red-teaming and safety-training efforts behind these models. Notably, new attacks utilizing our failure modes succeed on every prompt in a collection of unsafe requests from the models' red-teaming evaluation sets and outperform existing ad hoc jailbreaks. Our analysis emphasizes the need for safety-capability parity—that safety mechanisms should be as sophisticated as the underlying model—and argues against the idea that scaling alone can resolve these safety failure modes.

In recent months, large language models (LLMs) such as ChatGPT, Claude, and Bard have seen widespread deployment. These models exhibit advanced general capabilities [38], but also pose risks around misuse by bad actors (e.g., for misinformation or for crime [9, 32, 25, 30, 28]).

To mitigate these risks of misuse, model creators have implemented safety mechanisms to restrict model behavior to a "safe" subset of capabilities. These include both training-time interventions to align models with predefined values [41, 7] and post hoc flagging and filtering of inputs and outputs [56, 24, 52, 45]. These efforts are often complemented by red teaming, which proactively identifies and trains against weaknesses [42, 23, 38].

While hardening LLMs for safety can help [38], models remain vulnerable to adversarial inputs, as demonstrated by the spread of "jailbreaks" for ChatGPT on social media since its initial release [13, 17, 2]. These attacks are engineered to elicit behavior, such as producing harmful content or leaking personally identifiable information, that the model was trained to avoid. Attacks can range from elaborate role play (e.g., DAN [48]) to subtle subversion of the safety objective (see Figure 1(a)). Model creators have acknowledged and updated their models against jailbreak attacks [7, 38, 10, 5], but a systematic analysis and a conceptual understanding of this phenomenon remains lacking.

In this work, we analyze the vulnerability of safety-trained LLMs to jailbreak attacks by examining the model's pretraining and safety training processes. Based on known safety training methods, we hypothesize two failure modes—competing objectives and mismatched generalization—that shed

Preprint. Under review

Scalable Extraction of Training Data from (Production) Language Models

Milad Nasr*1 Nicholas Carlini*1 Jonathan Hayase^{1,2} Matthew Jagielski¹ A. Feder Cooper³ Daphne Ippolito^{1,4} Christopher A. Choquette-Choo¹ Eric Wallace⁵ Florian Tramèr⁶ Katherine Lee^{+1,3} ¹Google DeepMind ²University of Washington ³Cornell ⁴CMU ⁵UC Berkeley ⁶ETH Zurich *Equal contribution +Senior author

Abstract

This paper studies extractable memorization: training data that an adversary can efficiently extract by querying a machine learning model without prior knowledge of the training dataset. We show an adversary can extract gigabytes of training data from open-source language models like Pythia or GPT-Neo, semi-open models like LLaMA or Falcon, and closed models like ChatGPT. Existing techniques from the literature suffice to attack unaligned models; in order to attack the aligned ChatGPT, we develop a new divergence attack that causes the model to diverge from its chatbot-style generations and emit training data at a rate 150× higher than when behaving properly. Our methods show practical attacks can recover far more data than previously thought, and reveal that current alignment techniques do not eliminate memorization.

1 Introduction

Large language models (LLMs) memorize examples from their training datasets, which can allow an attacker to extract (potentially private) information [7, 12, 14]. Prior work has (a) performed large-scale studies of the total quantity of memorized training data for open-source models [11], and (b) developed practical attacks to extract training data on (relatively) small models like GPT-2, by manually annotating examples as memorized or not [14].

In this paper, we unify these two directions and perform a large-scale study of "extractable memorization" in language models. Unlike discoverable memorization [11] that captures an upper bound on all training data that is memorized (even if it can only be recovered by prompting the model with other training data), extractable memorization captures only that data that can be efficiently recovered by an adversary. We develop a scalable methodology that allows us to detect memorization in trillions of tokens of model outputs in terabytesized datasets, and perform this analysis on both open-source models (e.g., Pythia [5], GPT-Neo [6]) and semi-open models (e.g., LLaMA [49], Falcon [40]). We find that larger and more canable models are more vulnerable to data extraction attacks.

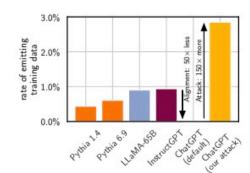


Figure 1: We scalably test for memorization in large language models. Models emit more memorized training data as they get larger. The aligned ChatGPT (gpt-3.5-turbo) appears 50× more private than any prior model, but we develop an attack that shows it is not. Using our attack, ChatGPT emits training data 150× more frequently than with prior attacks, and 3× more frequently than the base model.

But when we perform this analysis on gpt-3.5-turbo, it appears to memorize almost no training data. We hypothesize that this is because ChatGPT has been aligned (with RLHF [35, 37, 39, 44]) to act as a helpful chat assistant. 1

To circumvent the model's alignment, we discover a prompting strategy that causes gpt-3.5-turbo to "diverge" from reasonable, chatbot-style generations, and to behave like a base language model, outputting text in a typical Internet-text style. In order to check whether this emitted text was previously contained somewhere on the Internet, we merge together several publicly available web-scale training sets into a nine terabyte dataset. By matching against this dataset, we recover over ten thousand examples from ChatGPT's training dataset at a query cost of \$200 USD-and our scaling estimate suggests that one could extract over 10× more data with more queries.

¹While limited information is available about this model, similar models including those related to training data extraction [37, p. 13].

[C] Multi-step Jailbreaking Privacy Attacks on ChatGPT, Li et al, March 2023

[D] Jailbroken: How Does LLM Safety Training Fail?, Wei et al., Preprint, July 2023

[E] Scalable Extraction of Training Data from (Production) Language Models, Nasr et al., Preprint, November 2023

Privacy Attacks by Evading Safety Filters

Preventing Generation of Verbatim Memorization in Language Models Gives a False Sense of Privacy

Daphne Ippolito¹ Florian Tramèr*2 Milad Nasr*1

Chiyuan Zhang*¹ Matthew Jagielski*¹ Katherine Lee*1,3

Christopher A. Choquette-Choo*¹ Nicholas Carlini¹

¹ Google Research ² ETH Zurich ³ Cornell University

Abstract

Studying data memorization in neural language models helps us understand the risks (e.g., to privacy or copyright) associated with models regurgitating training data and aids in the development of countermeasures. Many prior works-and some recently deployed defenses-focus on "verbatim memorization", defined as a model generation that exactly matches a substring from the training set. We argue that verbatim memorization definitions are too restrictive and fail to capture more subtle forms of memorization. Specifically, we design and implement an efficient defense that perfectly prevents all verbatim memorization. And yet, we demonstrate that this "perfect" filter does not prevent the leakage of training data. Indeed, it is easily circumvented by plausible and minimally modified "style-transfer" prompts-and in some cases even the nonmodified original prompts-to extract memorized information. We conclude by discussing potential alternative definitions and why defining memorization is a difficult yet crucial open question for neural language models.

1 Introduction

The ability of neural language models to memorize their training data has been studied extensively (Kandpal et al., 2022; Lee et al., 2021; Carlini et al., 2022; Zhang et al., 2021; Thakkar et al., 2021; Ramaswamy et al., 2020). When language models, especially ones used in production systems, are susceptible to *data extraction* attacks, it can lead to practical problems ranging from privacy risks to copyright concerns. For example, Carlini et al. (2021) showed that the GPT-2 language model could output personally identifying information of individuals contained in the training dataset.

*Remaining authors ordered by Algorithm 18 in Appendix H; briefly, we require Daphne be listed first, and Nicholas listed last, and we search for the first permutation of authors' first names which satisfies these constraints, where permutations order names by their salted MD5 hash.



Figure 1: Illustration of Memorization-free Decoding, a defense which can eliminate verbatim memorization in the generations from a large neural language model, but does not prevent approximate memorization.

One natural way to avoid this risk is to filter out any generations which copy long strings verbatim from the training set. GitHub's Copilot, a languagemodel-based code assistant, deploys this defense by giving users the option to "block suggestions matching public code" (GitHub, 2022).

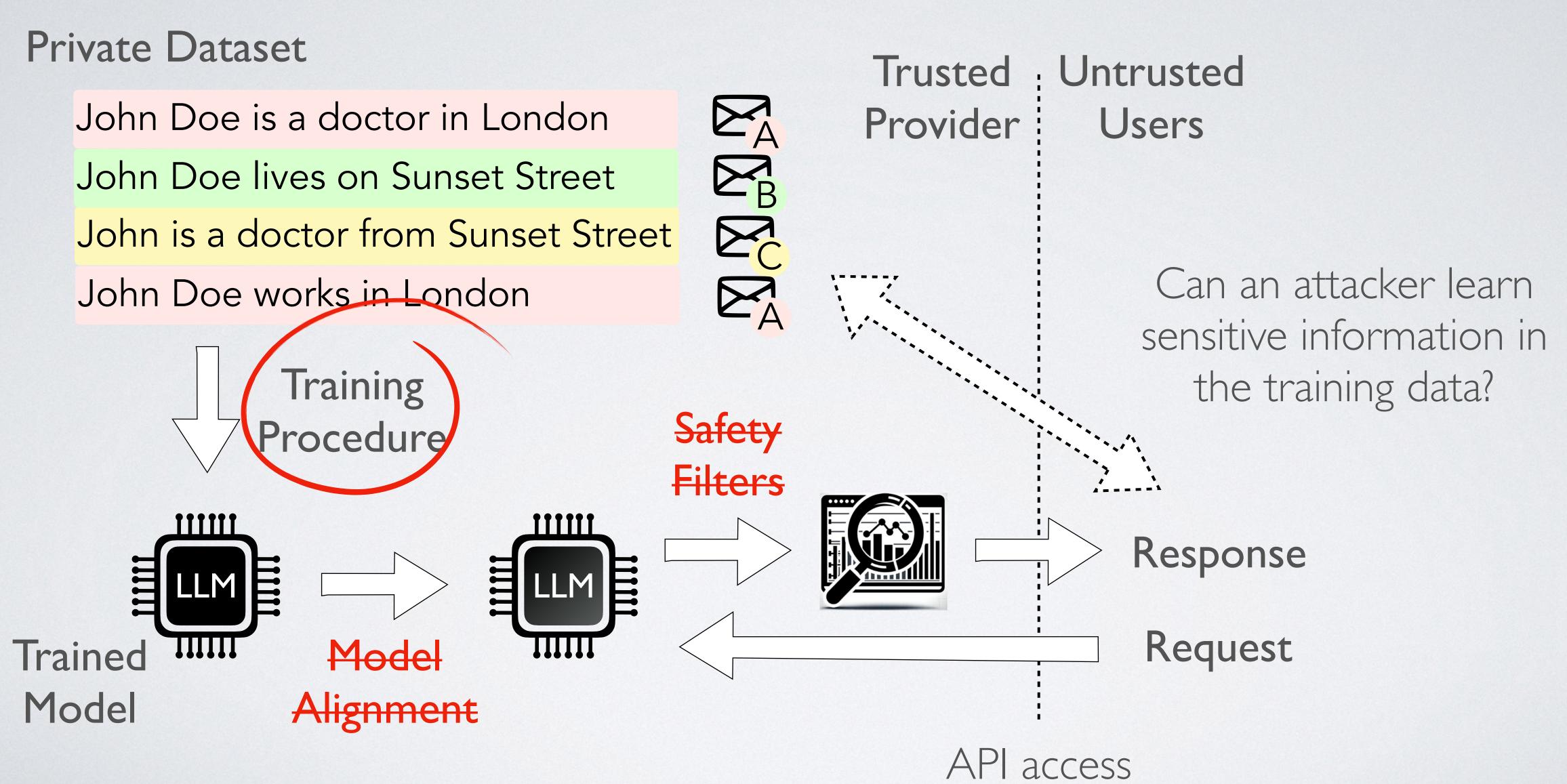
In this work, we ask the question: "Do language models emit paraphrased memorized content?" This scenario can happen maliciously (e.g., adversaries trying to extract private user data) or through honest interactions (e.g., users prompting in real-world scenarios). Indeed, we find that Copilot's filtering system is easy to circumvent by applying plausible "style transfers" to the prompt. For example, by translating variable names from English to French the model outputs completely memorized examples, but post-processed with the en-fr style transfer. We further show that GPT-3 (Brown et al., 2020), a model trained on natural language, is also vulnerable to extraction attacks.

Unfortunately, Copilot's training set and precise algorithm for their defense are non-public. Therefore, to investigate this phenomenon systematically, we develop MEMFREE decoding (Figure 1), an efficient defense that is guaranteed to prevent all verbatim memorization, and which scales to training sets consisting of hundreds of gigabytes of text. In

2

Proceedings of the 16th International Natural Language Generation Conference, pages 28-53 September 11-15, 2023. ©2023 Association for Computational Linguistics

Data Privacy for Large Language Models



Public Data and Private Information

What Does it Mean for a Language Model to Preserve Privacy?

Hannah Brown 1, Katherine Lee 2, Fatemehsadat Mireshghallah 3 Reza Shokri 1, Florian Tramèr 4*

¹National University of Singapore, ²Cornell University ³University of California San Diego, ⁴Google {hsbrown, reza}@comp.nus.edu.sg kate.lee168@gmail.com fatemeh@ucsd.edu tramer@google.com

Abstract

Natural language reflects our private lives and identities, making its privacy concerns as broad as those of real life. Language models lack the ability to understand the context and sensitivity of text, and tend to memorize phrases present in their training sets. An adversary can exploit this tendency to extract training data. Depending on the nature of the content and the context in which this data was collected, this could violate expectations of privacy. Thus, there is a growing interest in techniques for training language models that *preserve privacy*. In this paper, we discuss the mismatch between the narrow assumptions made by popular data protection techniques (data sanitization and differential privacy), and the broadness of natural language and of privacy as a social norm. We argue that existing protection methods cannot guarantee a generic and meaningful notion of privacy for language models. We conclude that language models should be trained on text data which was explicitly produced for public use.

1 Introduction

We use natural language to construct identities and communicate all our information in day-to-day life. Humans naturally understand when sharing a sensitive piece of information is appropriate based on context. It may be fine to share the same piece of information with one specific person or group, and a complete violation of privacy to share in another context, or at another point in time. Between humans, we trust that these implicit boundaries will be recognized and respected. As we build technologies that collect, store, and process our natural language communication, it is important that these technologies do not violate human notions of privacy or make use of data in ways beyond what is needed for the utility of the technology [71,101].

Language models (LMs) underlie much natural language technology we regularly interact with, from autocorrect to search engines and translation systems. Over the past few years, LMs have grown in size and now utilize unprecedentedly large datasets of natural language making privacy risks in LMs a far reaching problem. Prior work has already demonstrated that such models are prone to memorizing and regurgitating large portions of their training data [12, 13, 51, 38, 91]. Worse, they are especially likely to memorize atypical data points—which are more likely to represent privacy risks for the authors or subjects of these texts.

To address these privacy concerns, there is a growing body of literature that aims to create privacy-preserving language models [64, 2, 56, 98, 84, 40, 79]. While humans navigate the complexities of language and privacy by identifying appropriate contexts for sharing information, LMs are not currently designed to do this [14, 72, 66, 49, 66, 50, 41]. Instead, the approach to preserving privacy in LMs has been to attempt complete removal of private information from training data (data sanitization), or to design algorithms that do not memorize private data, such as algorithms that satisfy differential privacy (DP) [28, 26].

Both methods make explicit and implicit assumptions about the structure of data to be protected, the nature of private information, and requirements for privacy, that do not hold for the majority of natural language data. Sanitization techniques assume that private information can

- Data shared to intentionally violate someone's privacy (e.g., "doxing")
- Social media posts issued to a small target audience ("in-group sharing")
- Accidental leakage of other's information (e.g., conversations)

Privacy in Language Models [G]

^{*}Authors appear in alphabetical order

Base Model vs Fine-Tuning

Promises of Fine-Tuning [8]

- Improve Quality
- Steer Model
- Shorter Prompts
- Lower latency

Fine-tuning m	odels	Create your own custom models by fine-tuning our base models with your training data. Once you fine-tune a model, you'll be billed only for the tokens you use in requests to that model. Learn about fine-tuning >		
Model	Training	Input usage	Output usage	
gpt-3.5-turbo	\$0.0080 / 1K tokens	\$0.0030 / 1K tokens	\$0.0060 / 1K tokens	
davinci-002	\$0.0060 / 1K tokens	\$0.0120 / 1K tokens	\$0.0120 / 1K tokens	
babbage-002	\$0.0004 / 1K tokens	\$0.0016 / 1K tokens	\$0.0016 / 1K tokens	

OpenAl Pricing [7]

Focus of this Talk

Analyzing Leakage of Personally Identifiable Information in Language Models

Nils Lukas*§, Ahmed Salem†, Robert Sim†, Shruti Tople†, Lukas Wutschitz† and Santiago Zanella-Béguelin† *University of Waterloo, †Microsoft

nlukas@uwaterloo.ca, {t-salemahmed, rsim, shruti.tople, lukas.wutschitz, santiago}@microsoft.com

Abstract-Language Models (LMs) have been shown to leak information about training data through sentence-level membership inference and reconstruction attacks. Understanding the risk of LMs leaking Personally Identifiable Information (PII) has received less attention, which can be attributed to the false assumption that dataset curation techniques such as scrubbing are sufficient to prevent PII leakage. Scrubbing techniques reduce but do not prevent the risk of PII leakage: in practice scrubbing is imperfect and must balance the trade-off between minimizing disclosure and preserving the utility of the dataset. On the other hand, it is unclear to which extent algorithmic defenses such as differential privacy, designed to guarantee sentenceor user-level privacy, prevent PII disclosure. In this work, we introduce rigorous game-based definitions for three types of PII Fig. 1: An illustration of PII extraction, reconstruction and leakage via black-box extraction, inference, and reconstruction inference attack techniques. attacks with only API access to an LM. We empirically evaluate the attacks against GPT-2 models fine-tuned with and without defenses in three domains: case law, health care, and e-mails. Our main contributions are (i) novel attacks that can extract up to $10\times$ more PII sequences than existing attacks, (ii) showing that sentence-level differential privacy reduces the risk of PII disclosure but still leaks about 3% of PII sequences, and (iii) a subtle reconstruction. Code to reproduce all experiments in the paper is available at https://github.com/microsoft/analysing_pii_leakage.

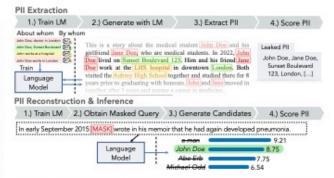
I. INTRODUCTION

Language Models (LMs) are fundamental to many natural language processing tasks [22, 49]. State-of-the-art LMs scale to trillions of parameters [19] and are pre-trained on large text utility that does not leak PII and make it available as a blackcorpora (e.g., 700GB [53]). Pre-trained LMs are adapted to box API. The threat is an attacker who learns PII, such as downstream tasks by fine-tuning on domain-specific datasets such as human dialogs [7] or clinical health data [62] which LM. Extracting any PII by itself, such as a personal address, may contain private information.

and about whom it contains information, known as data extraction [9, 11, 29, 59, 69]. These two categories can be disjoint but associations in the latter can be used to infer information about the former. For LMs, data extraction is a significant threat in practice since attackers with black-box API access can extract at least 1% of the training data [11]. Existing work focuses on finding a lower bound on any kind

of memorization but does not differentiate public and private

[§]Part of this work was done during an internship at Microsoft Research. *To cite this work, please refer to the full publication [41] in IEEE Security



leaked information. For example, leaking highly duplicated common phrases is not a privacy violation according to the GDPR [17] as opposed to leaking Personally Identifiable Information (PII). In practice, any LM trained on real, sensitive data has to protect PII, but memorization of PII is not well understood. We believe that a comprehensive study on the risk of PII memorization in LMs is missing.

Consider a service provider who wants to deploy a nextword prediction LM for composing e-mails, such as Google's Smart Compose [13]. Their goal is to train an LM with high names, addresses or other sensitive information through the can already pose a privacy threat. This threat is elevated Memorization is a privacy concern in LMs [9]. The threat when an attacker can associate a piece of PII to a context, is that an attacker learns by whom the training data was for example, "In May 2022, [MASK] had chemotherapy at provided, known as membership inference [30, 45, 46, 58] LHS". As a part of this paper, we study the feasibility of such attacks on LMs in practice. Figure 1 illustrates the type of PII

> Defenses against memorization are based on dataset curation and algorithmic defenses. PII scrubbing is a dataset curation technique that removes PII from text, relying on Named Entity Recognition (NER) [35] to tag PII. Modern NER is based on the Transformer architecture [63] and has mixed recall of 97% (for names) and 80% (for care unit numbers) on clinical health data, meaning that much PII is retained after scrubbing [62]. Machine learning pipelines incorporate algorithmic defenses such as differentially-private







Robert Sim







Lukas Wutschitz



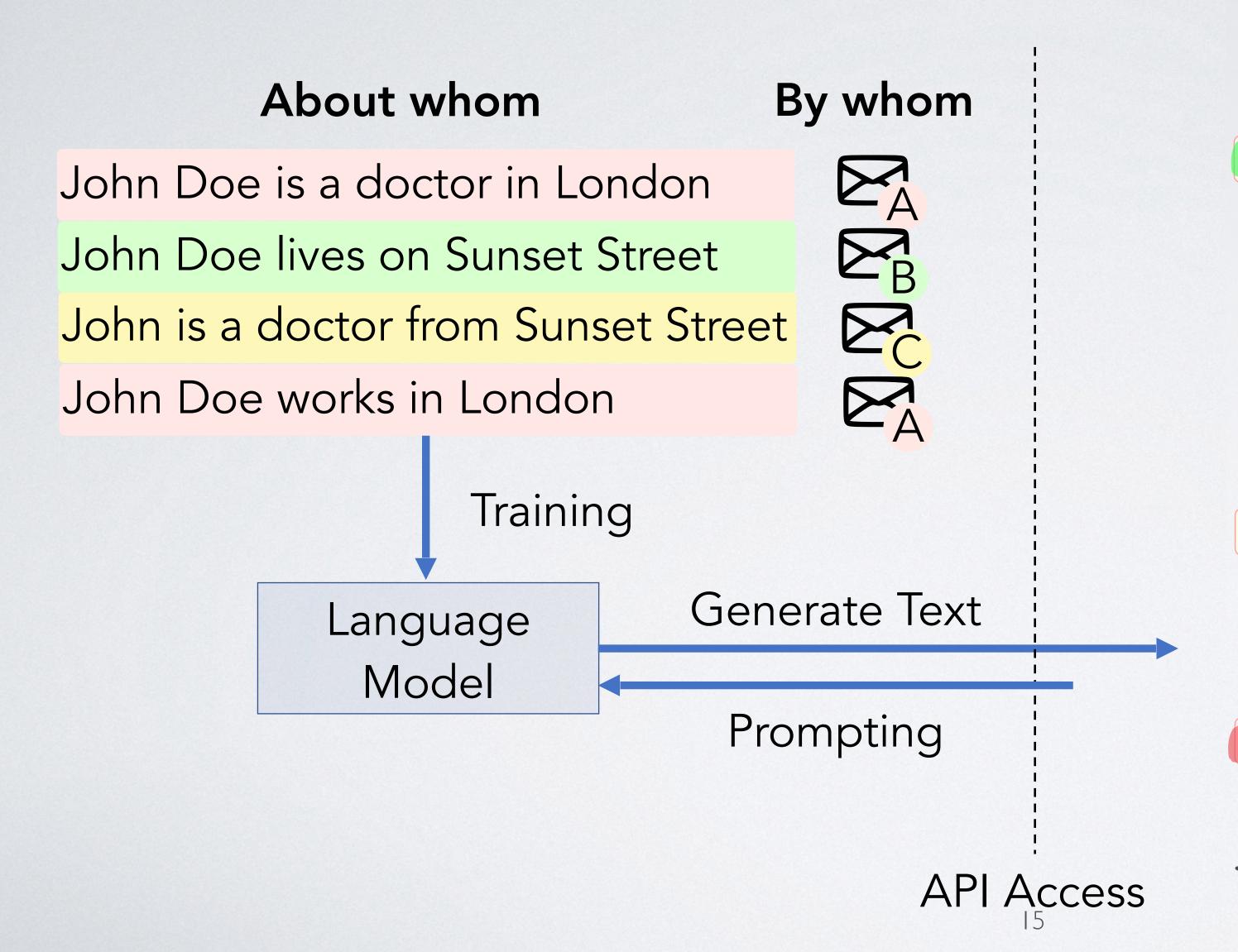
Santiago # Zanella-Béguelin

[H] Analyzing Leakage of Personally Identifiable Information in Language Models, Lukas et al., February 2023









Once upon a time, there existed a tale of medical students John Doe and his girlfriend, Jane Doe. In the year 2022, John resided at Sunset Street while pursuing his medical education. Alongside his friend Jane, he worked at the LHS Hospital located in the bustling heart of downtown London. Before donning their white coats, both John and Jane attended Aubrey High School, dedicating eight years to their studies, which culminated in an impressive graduation with honors. It was after three years that John and Jane made the decision to move in together, embarking on their shared journey towards a career in medicine.

1.) PII Extraction

Once upon a time, there existed a tale of medical students John Doe and his girlfriend, Jane Doe. In the year 2022, John resided at Sunset Street while pursuing his medical education. Alongside his friend Jane, he worked at the LHS Hospital located in the bustling heart of downtown London. Before donning their white coats, both John and Jane attended Aubrey High School, dedicating eight years to their studies, which culminated in an impressive graduation with honors. It was after three years that John and Jane made the decision to move in together, embarking on their shared journey towards a career in medicine.

1.) PII Extraction

John Doe London Sunset Street LHS Hospital Jane Doe
Aubrey High School

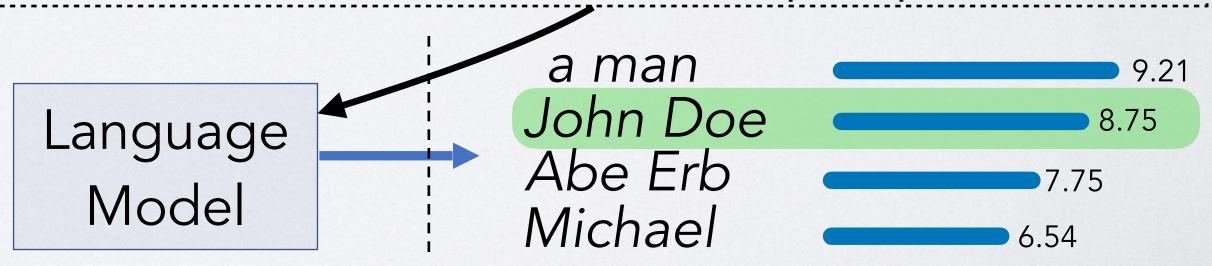
Real

2.) PII Reconstruction & 3.) PII Inference

Fictional

Real Sentence

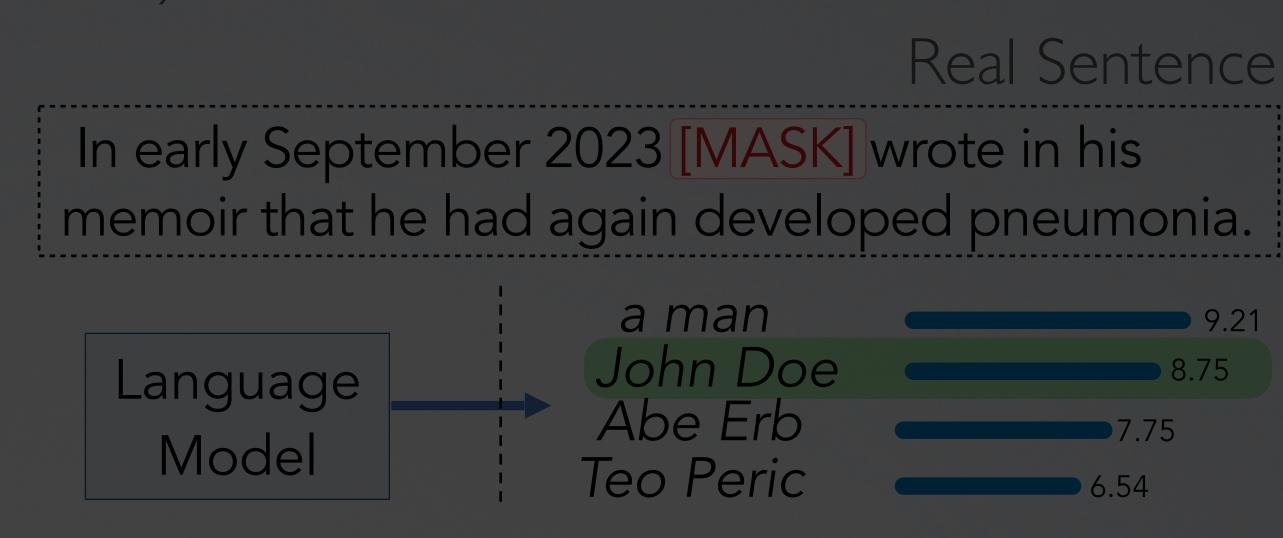
In early September 2023 [MASK] wrote in his memoir that he had again developed pneumonia.

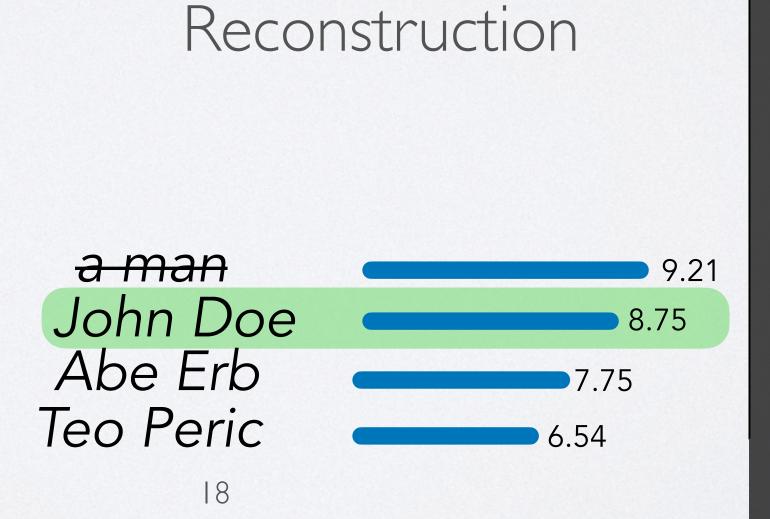


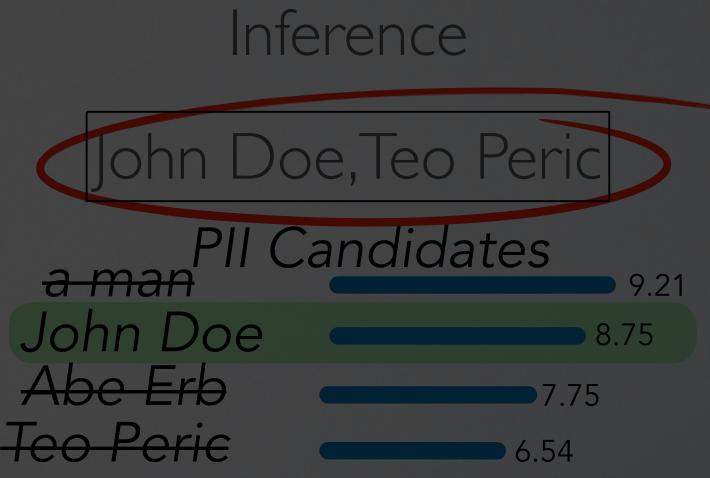
Once upon a time, there existed a tale of medical students John Doe and his girlfriend, Jane Doe. In the year 2022, John resided at Sunset Street while pursuing his medical education. Alongside his friend Jane, he worked at the LHS Hospital located in the bustling heart of downtown London. Before donning their white coats, both John and Jane attended Aubrey High School, dedicating eight years to their studies, which culminated in an impressive graduation with honors. It was after three years that John and made the decision to move in together, embarking on their shared journey towards a career in medicine.

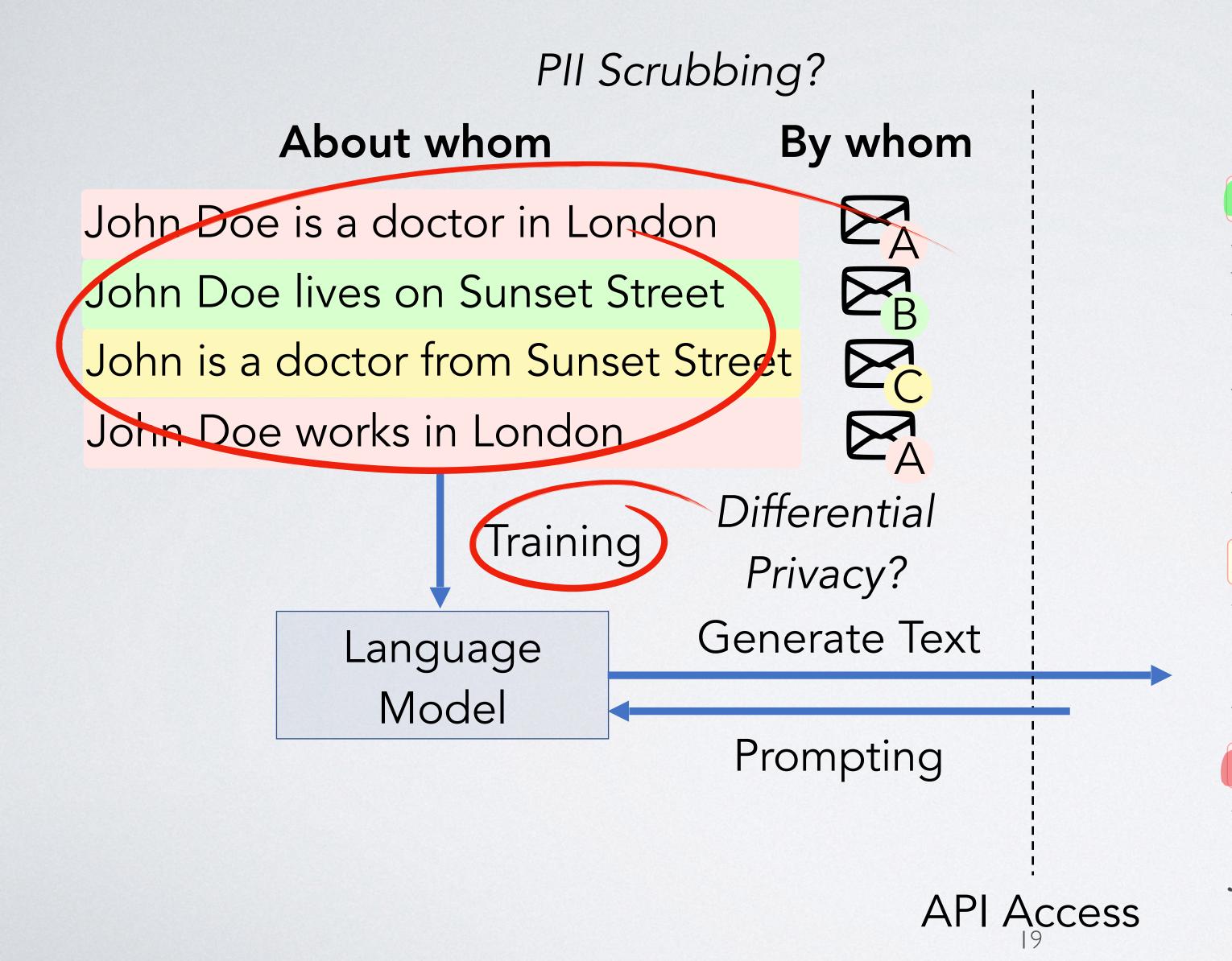
Motivation

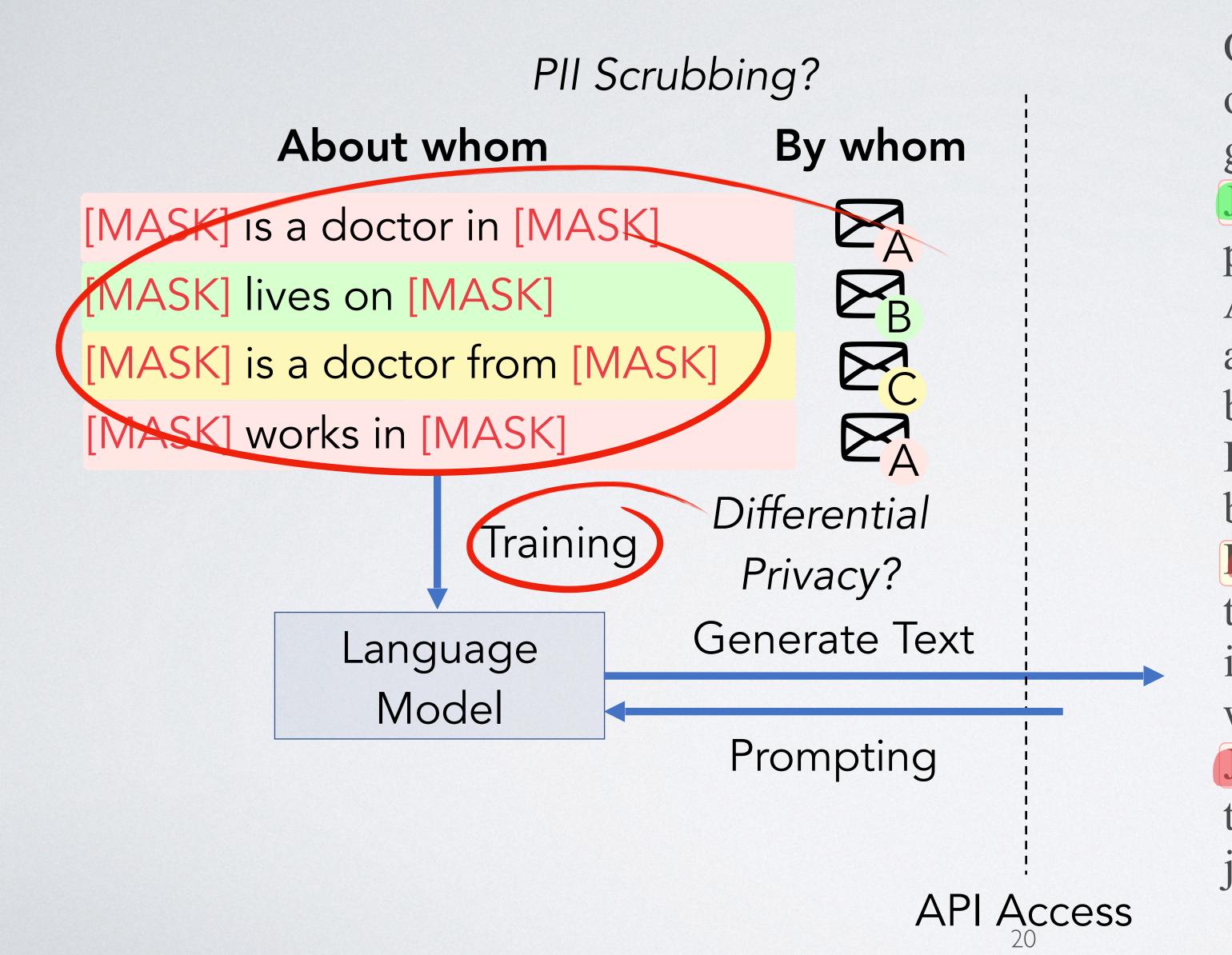
2.) PII Reconstruction & Inference

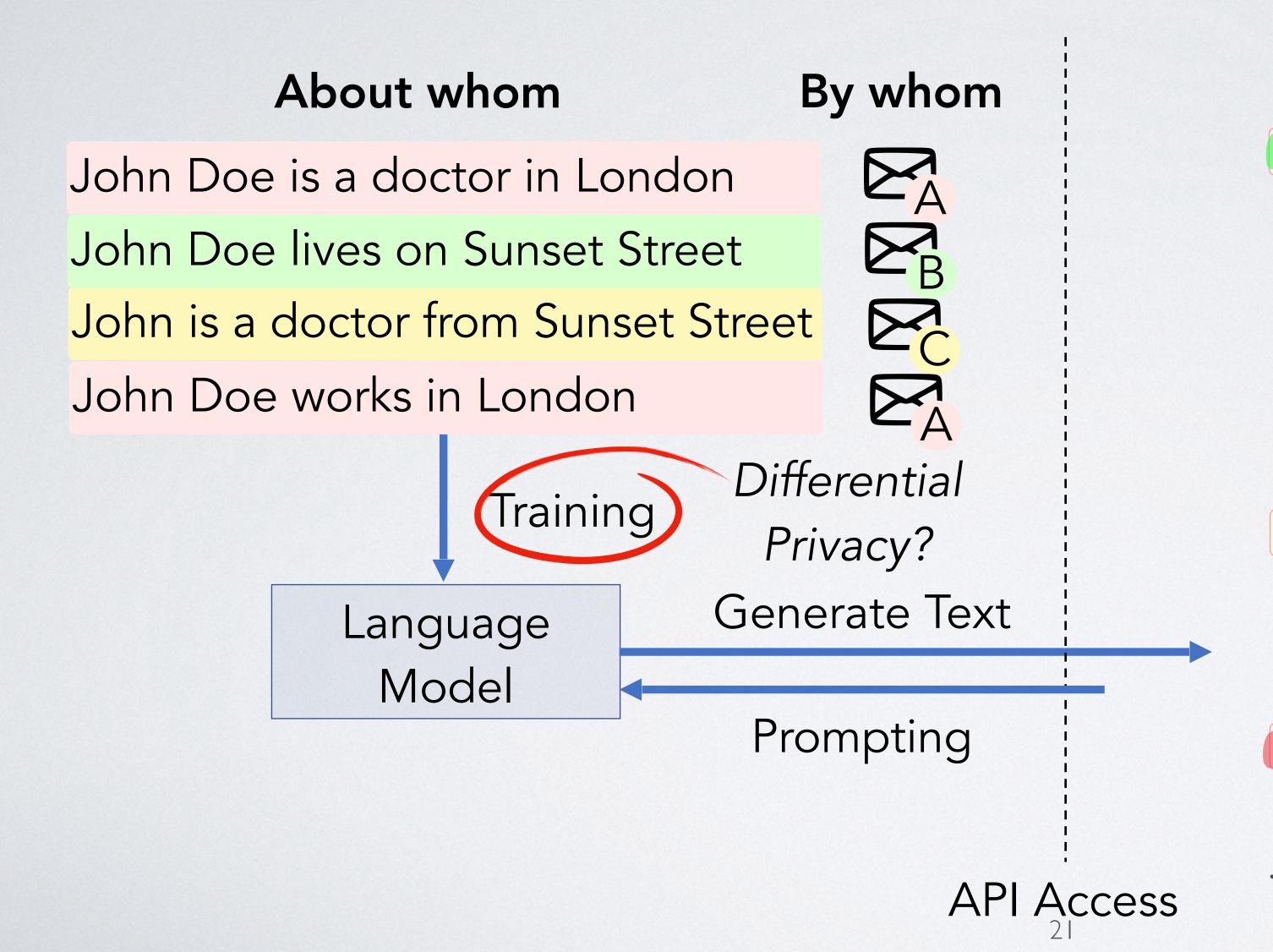




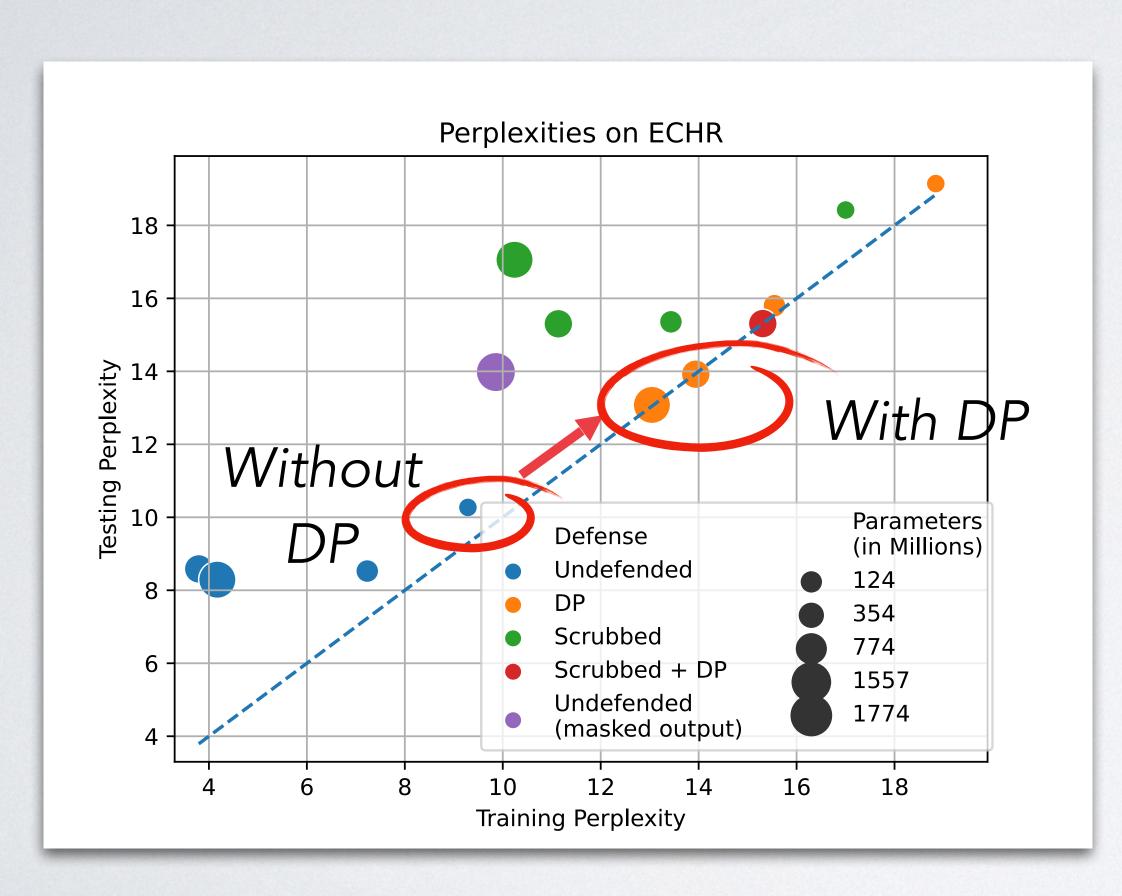




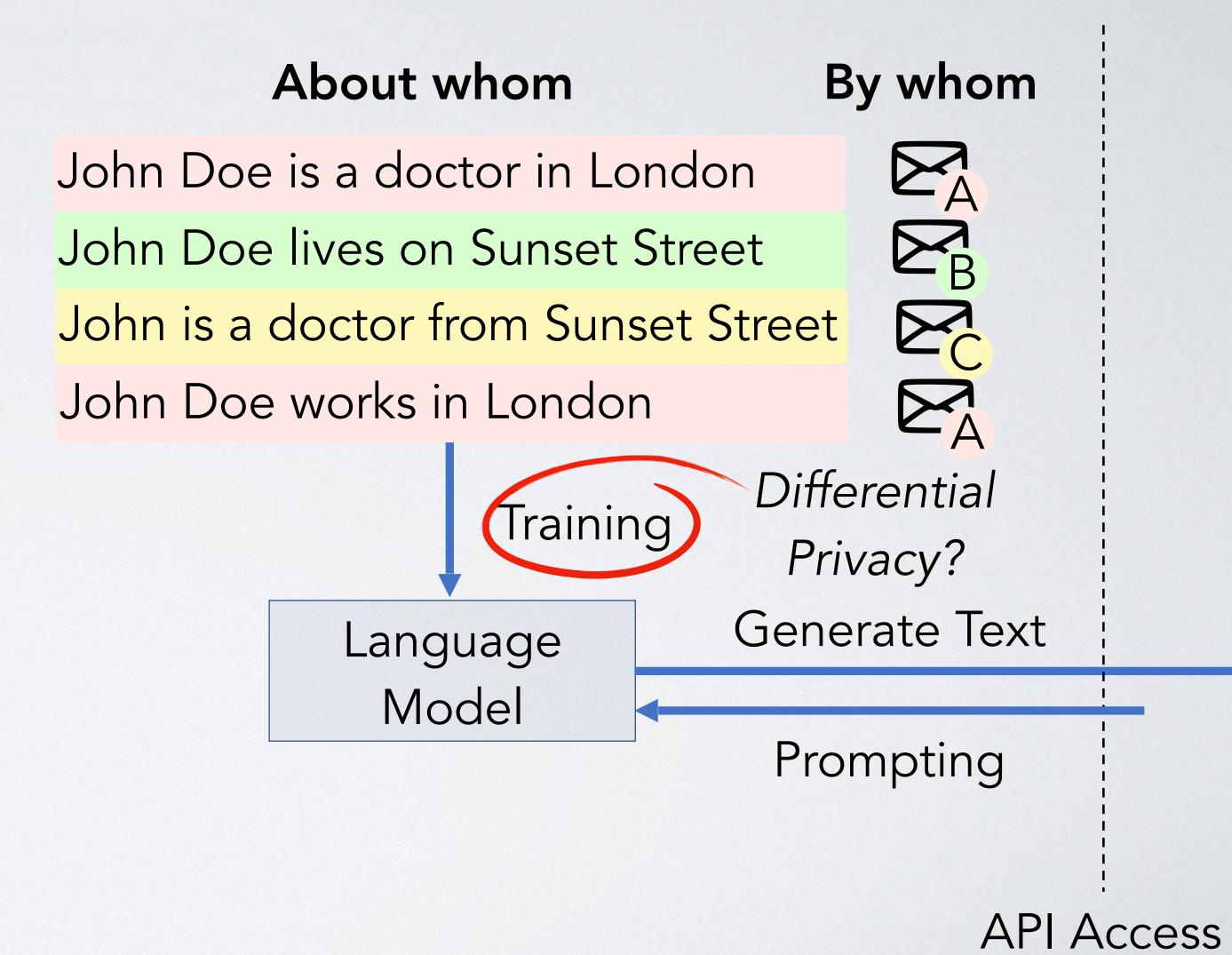




Problems with Differential Privacy

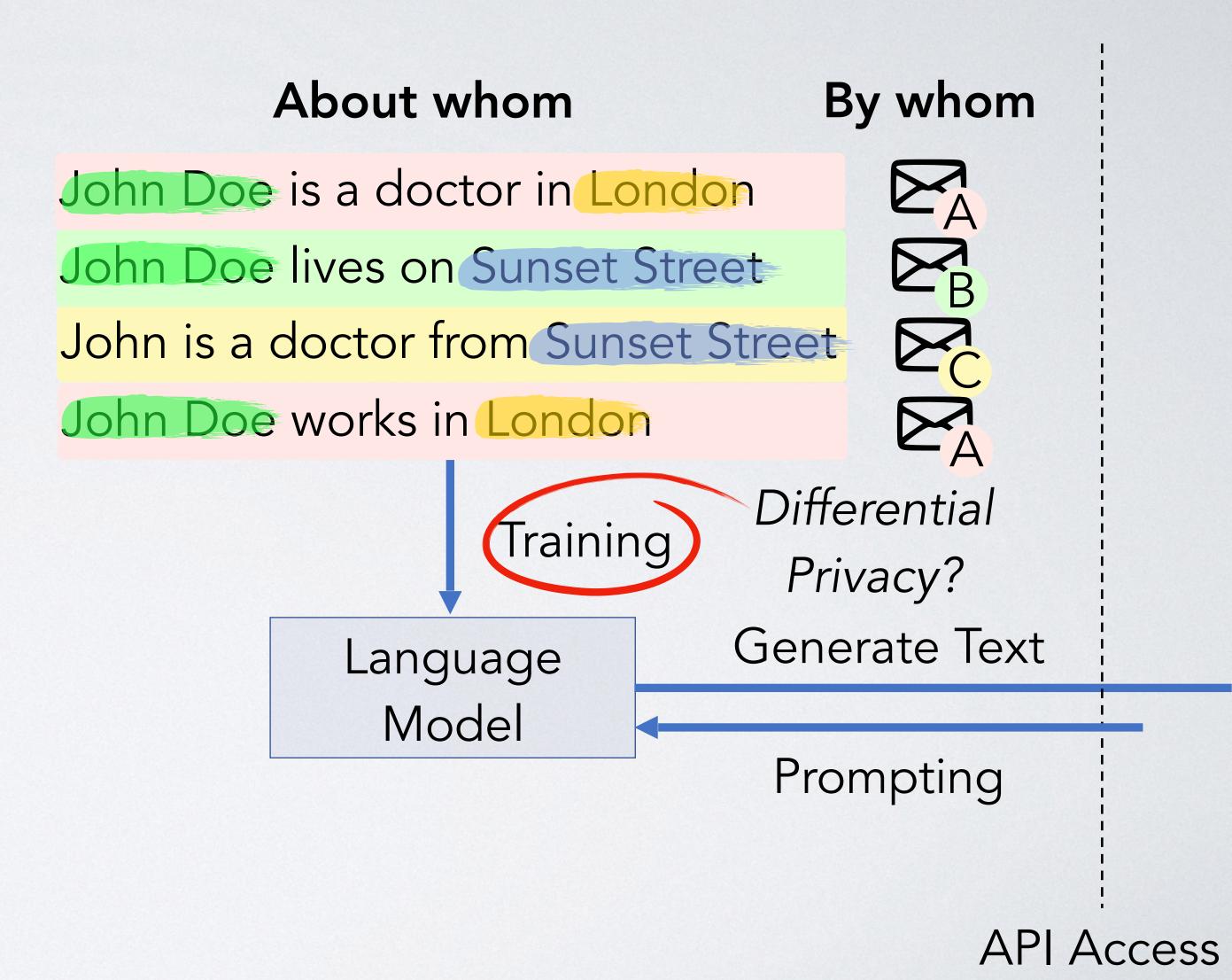


Privacy at the cost of Model Utility



Problems with Differential Privacy

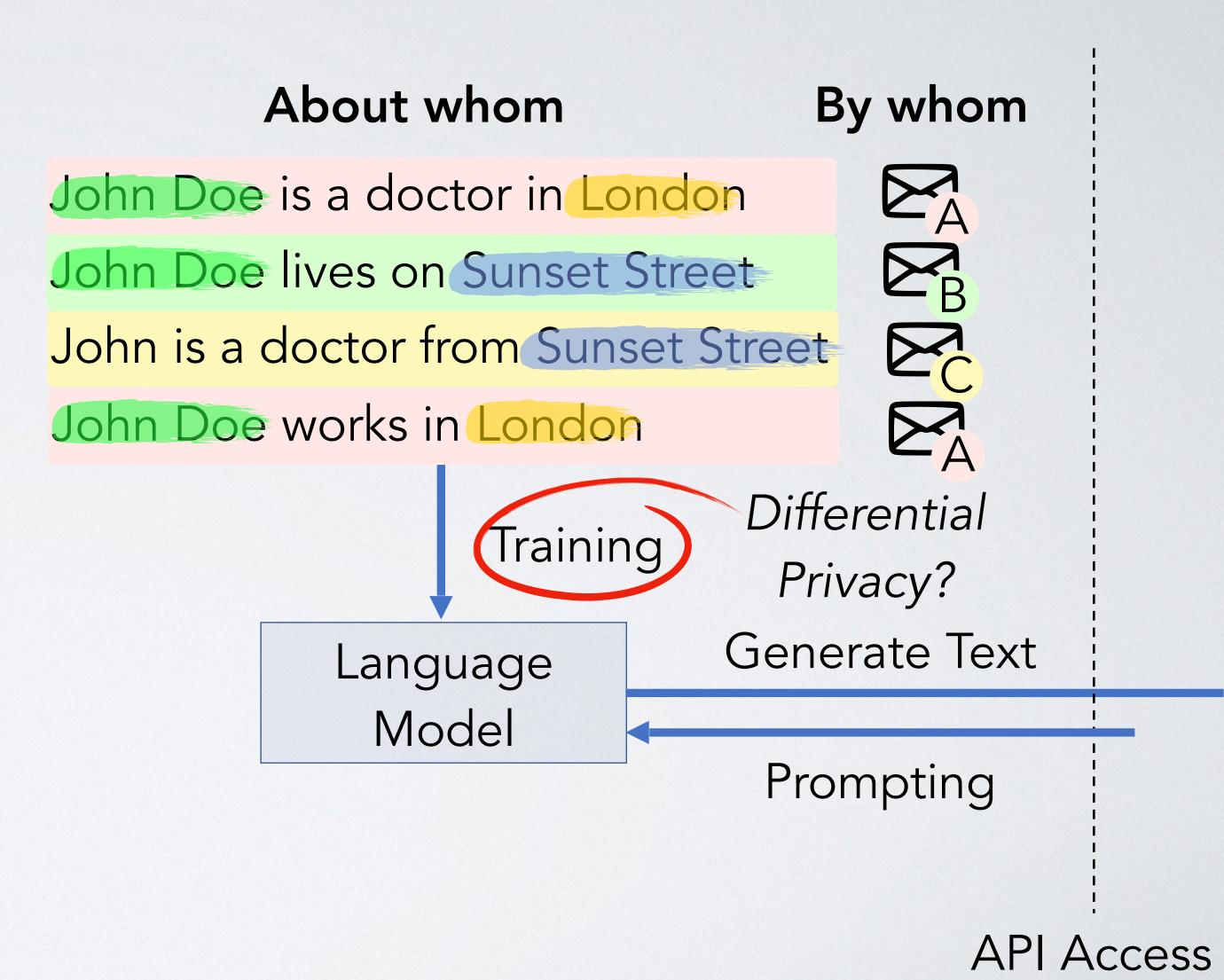
DP protects against an attacker learning by whom data was provided, but not about whom it contains information.

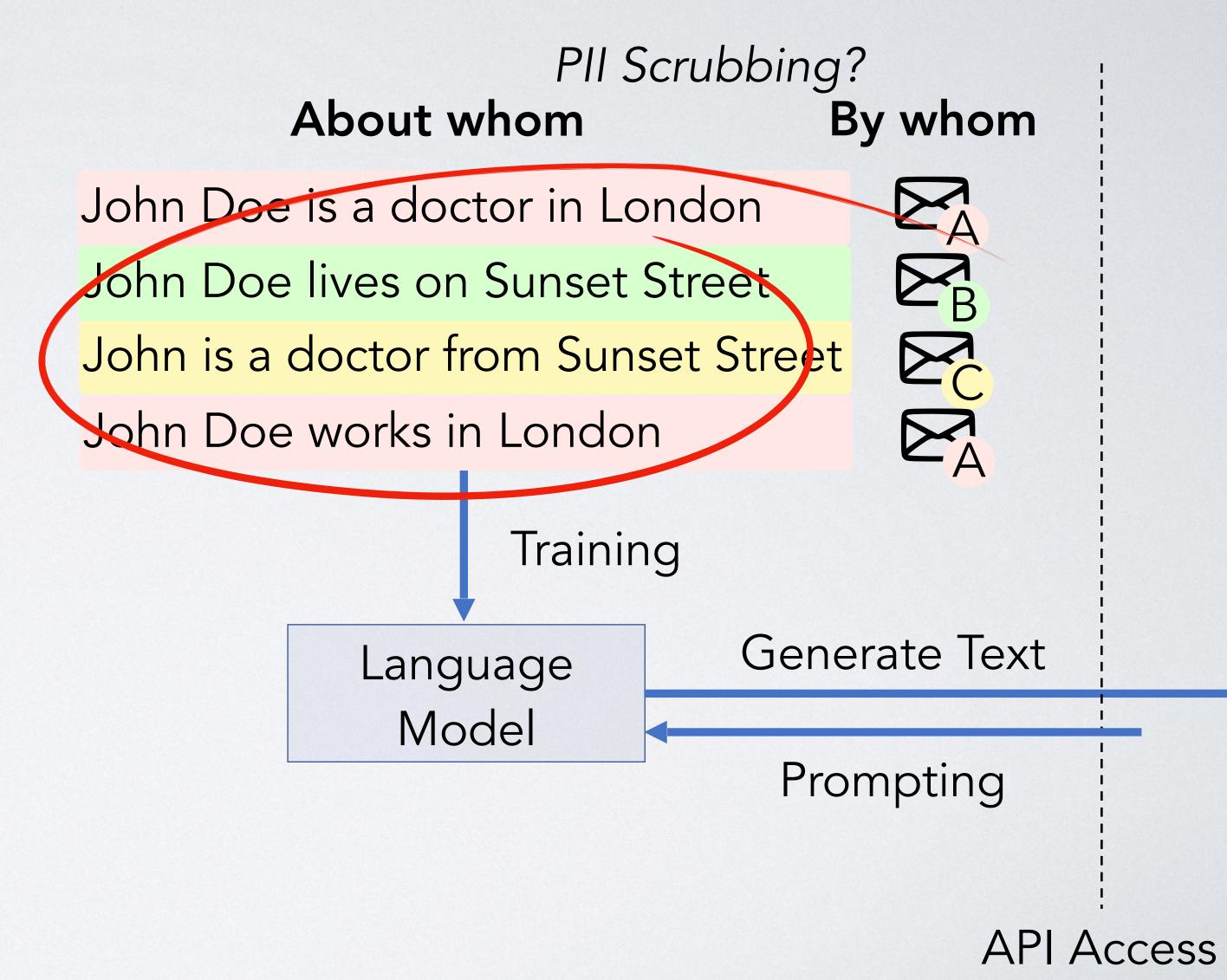


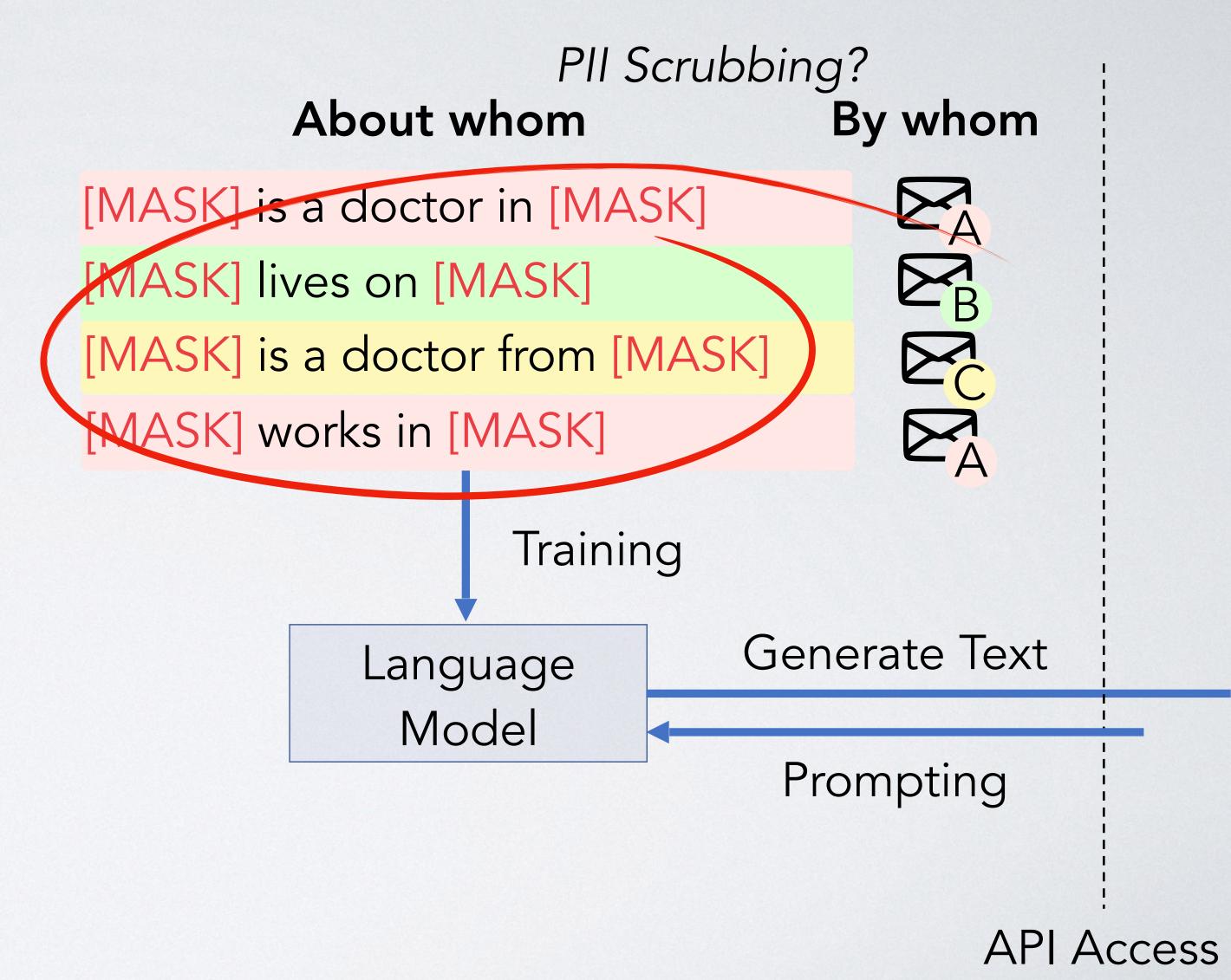
Problems with Differential Privacy

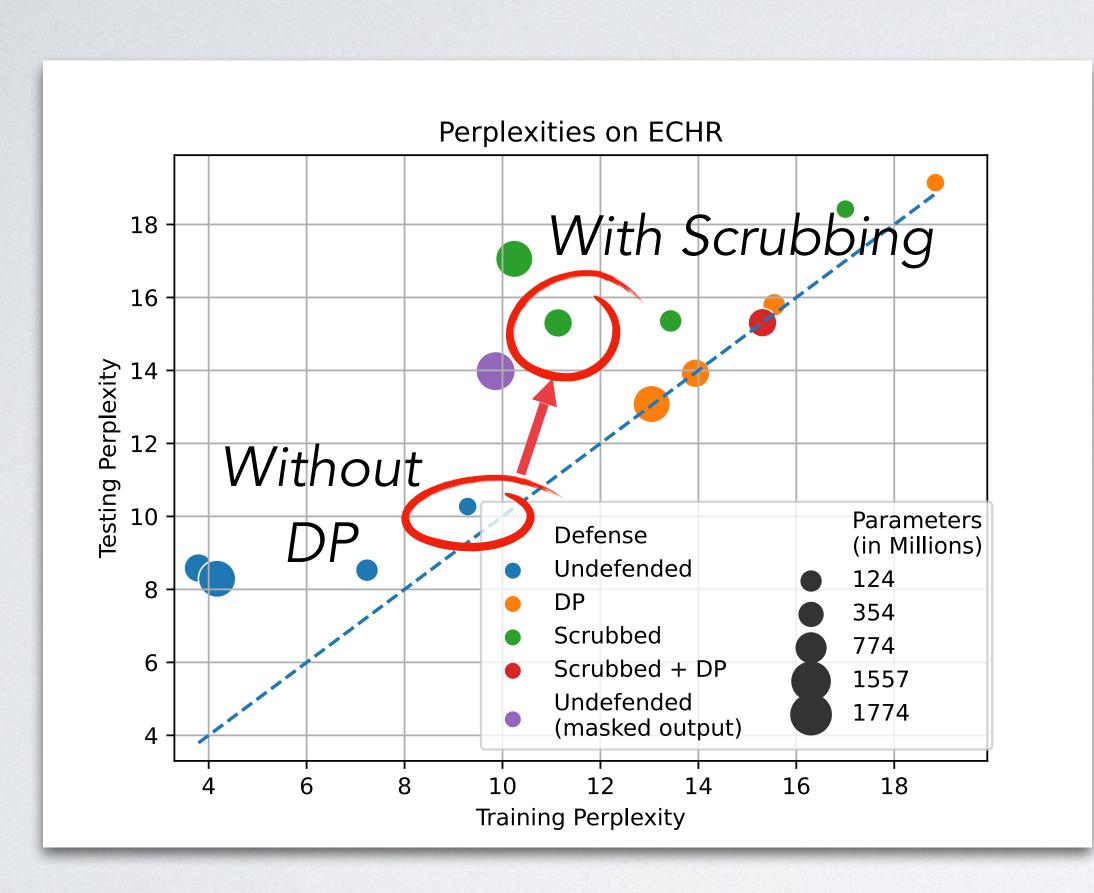
Group-level DP can help but ..

- 1) Group sizes are not always known a priori and under worst-case assumptions has deleterious impact on model utility.
- 2) PII Duplication across groups

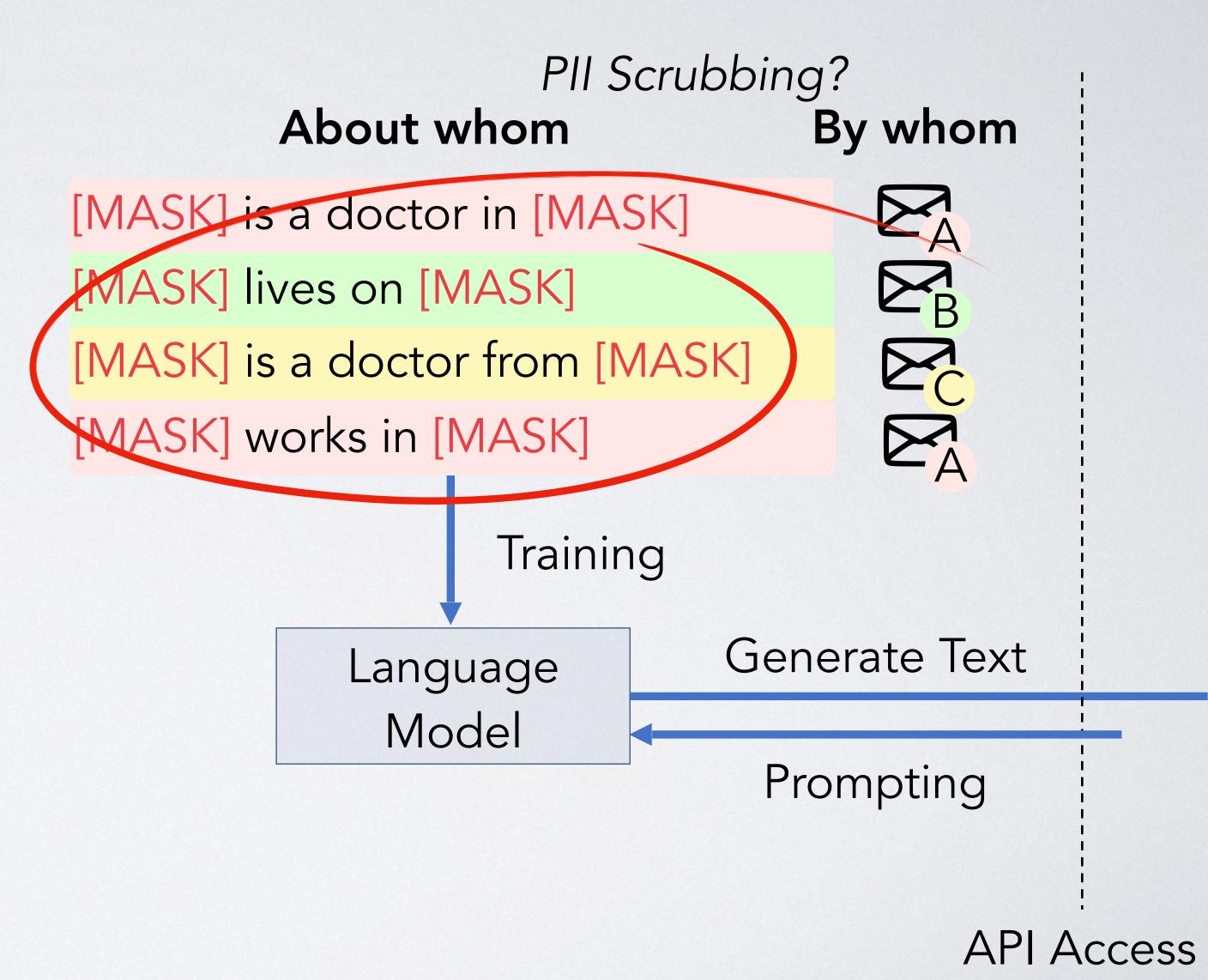




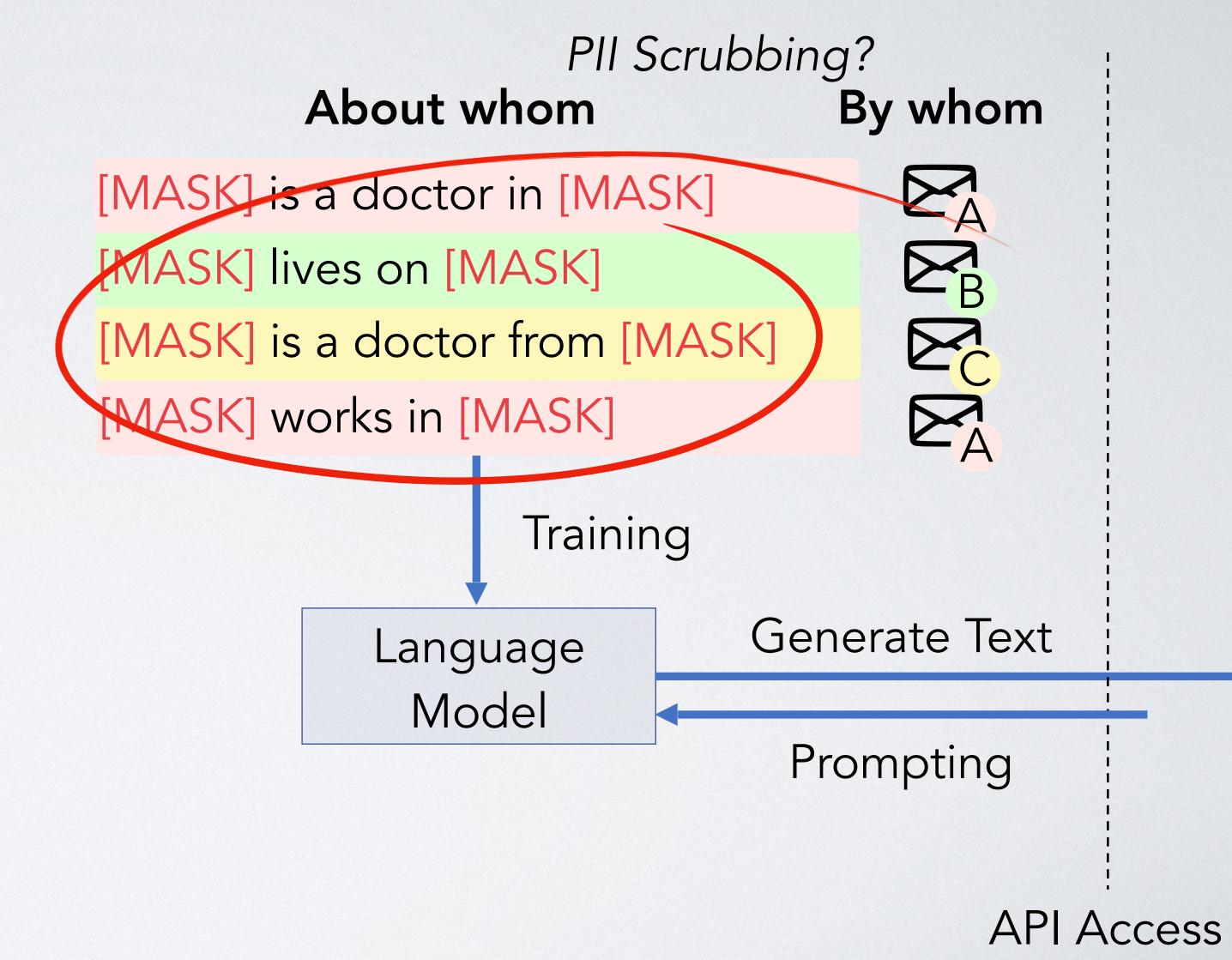




Privacy at the cost of Model Utility



Methods to optimize the privacy/utility trade-off are missing.



Related Work

Canaries

The Secret Sharer: Evaluating and Testing **Unintended Memorization in Neural Networks**

Nicholas Carlini^{1,2} Chang Liu² Úlfar Erlingsson¹ Jernej Kos³ Dawn Song²

This paper describes a testing methodology for quantitatively assessing the risk that rare or unique training-data sequences are unintentionally memorized by generative sequence models—a common type of machine-learning model. Because such models are sometimes trained on sensitive data

Because such models are sometimes trained on sensitive data (e.g., the text of users' private messages), this methodology can benefit privacy by allowing deep-learning practitioners to select means of training that minimize such memorization. In experiments, we show that unintended memorization is a persistent, hard-to-avoid issue that can have serious conse-quences. Specifically, for models trained without considera-tion of memorization, we describe new, efficient procedures that can extract unioue, secret sequences, such as credit card numbers. We show that our testing strategy is a practical and easy-to-use first line of defense, e.g., by describing its application to quantitatively limit data exposure in Google's

When a secret is shared, it can be very difficult to prevent its further disclosure—as artfully explored in Joseph Conrad's The Secret Sharer [10]. This difficulty also arises in machineand models' training involves sensitive or private data.

Disclosure of secrets is of particular concern in neural-

language text. First, such text will often contain sensitive or private sequences, accidentally, even if the text is supposedly public. Second, such models are designed to learn text patterns such as grammar, turns of phrase, and spelling, which

terms such as grammar, turns of phrase, and spelling, which comprise a vanishing fraction of the exponential space of all possible sequences. Therefore, even if sensitive or private training-data text is very rare, one should assume that well-trained models have paid attention to its precise details. Concretely, disclosure of secrets may arise naturally in generative text models like those used for text auto-completion and predictive keyboards, if trained on possibly-sensitive data. The users of such models may discover—either by accident or on purpose—that entering certain text prefixes causes the models to output surprisingly-revealing text completions [37].

¹Google Brain ²University of California, Berkeley ³National University of Singapore

For example, users may find that the input "my social-security number is..." gets auto-completed to an obvious secret (such as a valid-looking SSN not their own), or find that other inputs are auto-completed to text with oddly-specific details. So triggered, unscrupulous or curious users may start to "attack" such models by entering different input prefixes to try to min possibly-secret suffixes. Therefore, for generative text mod els, assessing and reducing the chances that secrets may be disclosed in this manner is a key practical concern.

To enable practitioners to measure their models' propensit handouces a quantitative metric of exposure. In its metric to a be applied during training as part of a testing methodology that empirically measures a model's potential for unintended memorization of unique or rare sequences in the training data.

Our exposure metric conservatively characterizes knowlby accident (or by a most-likely beam search). As validation of this, we describe an algorithm guided by the exposure metric that, given a pretrained model, can efficiently extract secret sequences even when the model considers parts of them to be highly unlikely. We demonstrate our algorithm's effectiveness in experiments, e.g., by extracting credit card numbers from a horsest and the size of the properties of the propert language model trained on the Enron email data. Such empir al extraction has proven useful in convincing practition that unintended memorization is an issue of serious, practical concern, and not just of academic interest.

Our exposure-based testing strategy is practical, as we demonstrate in experiments, and by describing its use in removing privacy risks for Google's Smart Compose, a deemail messages and used by other users for predictive text

In evaluating our exposure metric, we find unintended me particular, such memorization is not due to overtraining [47]:
it occurs early during training, and persists across different
types of models and training strategies—even when the memorized data is very rare and the model size is much smaller
than the size of the training data corpus. Furthermore, we
show that simple, intuitive regularization approaches such
as early-stopping and dropout are insufficient to prevent unintended memorization. Only by using differentially-private
training techniques are we able to eliminate the issue completely, albeit at some loss in utility. N-grams

Sequences

How much do language models copy from their training data:

R. Thomas McCoy,* 1 Paul Smolensky,2,1 Tal Linzen,3 Jianfeng Gao,2 Asli Celikyilmaz* ¹Johns Hopkins University ²Microsoft Research ³New York University ⁴Facebook AI Research tom.mccoy@jhu.edu, psmo@microsoft.com, linzen@nyu.edu, jfgao@microsoft.com, aslic@fb.com

they have seen before, or have they learned generalizable linguistic abstractions? To duce RAVEN, a suite of analyses for as sessing the novelty of generated text, fo-cusing on sequential structure (n-grams) and GPT-2). For local structure-e.g., indi is substantially less novel than our base line of human-generated text from each model's test set. For larger-scale structure—

generated text is as novel or even more novel than the human-generated baseline over 1,000 words long from the training se showing that GPT-2's novel text is usuall well-formed morphologically and syntacti cally but has reasonably frequent semantic issues (e.g., being self-contradictory).

"discovering intricate structures" that support sopects of structure that we analyze: they generphisticated generalization (LeCun et al., 2015), or are they "stochastic parrots" that simply memo-rize seen examples and recombine them in shallow ways (Bender et al., 2021)?

models (LMs) can generate grammatical, coherent text (See et al., 2019; Brown et al., 2020, section

instructed by the model or copied from the train ing set. We argue that it is important to disentangle these possibilities. That is, in addition to evaluat ing the *quality* of generated text, as is already stan-dard (Gatt and Krahmer, 2018; Celikyilmaz et al.,

2020), we should also evaluate its novelty. Novelty is important for several reasons. From a linguistic perspective, one core component of knowing a language is the ability to combine familiar parts in novel ways (Chomsky, 1957; Hockett, 1963). From a machine learning perspective, models are meant to learn the training distribu-1995). Finally, on the more practical side, models that copy training data might leak sensitive information (Carlini et al., 2021) or repeat hate speech (Bender et al., 2021).

In this work, to assess the novelty of gener ated text, we introduce a suite of analyses called RAVEN (RAting VErbal Novelty). 1.2 These analyses cover both sequential structure (n-grams) and syntactic structure. We apply these analy ses to text generated by an LSTM, a Transformer, Transformer-XL, and all 4 sizes of GPT-2 (the largest LM for which we had access to the training data). Because there are many ways to generat text from LMs, we test 12 generation methods and 4 prompt lengths. As a baseline, we also analyze human-generated text from each model's test set. We find that models display novelty for all as-

are they "stochastic parrots" that simply memorize seen examples and recombine them in shallow
ways (Bender et al., 2021)?

We focus on this question in the area of openended text generation. Neural network language

"Neural network language"

"GitHub code will be released soon.

"GitHub code will be released soon.

"Werhal here uses its broad definition of "tinguistic" rather
than the narrow definition of "verb-related." This acronym
refers to "The Raven" by Edgar Allan Poe, in which the narratranscended that the properties of the properti

Extracting Training Data from Large Language Models

Eric Wallace3 Ariel Herbert-Voss^{5,6} Katherine Lee¹ Adam Roberts¹ Tom Brown⁵ Dawn Song³ Úlfar Erlingsson⁷ Alina Oprea⁴ Colin Raffel¹ ¹Google ²Stanford ³UC Berkeley ⁴Northeastern University ⁵OpenAI ⁶Harvard ⁷Apple

It has become common to publish large (billion parameter) language models that have been trained on private datasets. This paper demonstrates that in such settings, an adversary can perform a training data extraction attack to recover individual

trained on scrapes of the public Internet, and are able to extrac hundreds of verbatim text sequences from the model's training data. These extracted examples include (public) personal addresses), IRC conversations, code, and 128-bit UUIDs. Ou attack is possible even though each of the above sequences are included in just one document in the training data

Language models (LMs)-statistical models which assign a tasks [29, 39, 55], even without updating their parameters [7]. At the same time, machine learning models are notorious for exposing information about their (potentially private) training data—both in general [47,65] and in the specific case of language models [8,45]. For instance, for certain models it is known that adversaries can apply membership inference attacks [65] to predict whether or not any particular example

[75]—when a model's training error is significantly lower than its test error-because overfitting often indicates that a model has memorized examples from its training set. Indeed overfitting is a sufficient condition for privacy leakage [72] and many attacks work by exploiting overfitting [65]

son's name, email address, phone number, fax number, and

sysical address. The example in this figure shows inform

East Stroudsburg Stroudsburg...

GPT-2

The association between overfitting and memorization has— erroneously—led many to assume that state-of-the-art LMs will *not* leak information about their training data. Because these models are often trained on massive de-duplicated datasets only for a single epoch [7, 55], they exhibit little to no overfitting [53]. Accordingly, the prevailing wisdom has been that "the degree of copying with respect to any given work is likely to be, at most, de minimis" [71] and that models

USENIX Association

30th USENIX Security Symposium 2633

McCoy et al., 2019 Carlini et al., 2019

Carlini et al., 2020

ABSTRACT and hurts fairness (some texts are memorized over others) We describe three log-linear relationships that quantify the degree to which LMs emit memorized training data. Memorization significantly grows as we increase (1 the capacity of a model, (2) the number of times an example has been duplicated the capacity of a model, (2) the number of times an example has been duplicated, and (3) the number of tokens of context used to prompt the model. Surprisingly, we find the situation becomes more complicated when generalizing these results across model families. On the whole, we find that memorization in LMs is more prevalent than previously believed and will likely get worse as models continues to scale, at least without active mitigations. 1 Introduction The performance of neural language models has continuously improved as these models have grown from millions to trillions of parameters (Fedus et al.] [2021], with their training sets similarly growing from millions to trillions of tokens. In anticipation of future, even larger models trained on minimally curated datasets, it is important to quantify factors that lead to increased memorization of a model's In addition to prior work's loose estimates of models' memorization capabilities, there is a limited understanding of how memorization varies across different neural language models and datasets of different scales. Frior studies of memorization in language models either focus on models of datasets of a fixed size (Carlini et al., 2019; Zhang et al., 2021; Thakkar et al., 2020) or identify a narrow memorization-versus-scale relationship (Carlini et al., 2020). [Lee et al., 2021]. While [McCoy] et al., 2021] broadly study the extent to which language models memorize, their focus is on how to avoid the problem and ensure novelty of model outputs, rather than on studying model risk through identifying the maximal amount of data memorization. identifying the maximal amount of data memorization

QUANTIFYING MEMORIZATION ACROSS

NEURAL LANGUAGE MODELS

Carlini et al., 2022

PII Leakage In Pre-Trained LMs

Are Large Pre-Trained Language Models Leaking Your Personal Information?

Jie Huang* Hanyin Shao* Kevin Chen-Chuan Chang {jeffhj, hanvins2, kcchang}@illinois.edu

Are Large Pre-Trained Language Models Leaking Your Personal Information? In this paper, we analyze whether Pre-Trained Lan-guage Models (PLMs) are prone to leaking personal information. Specifically, we query PLMs for email addresses with contexts of the email addresses with contexts of the email address or prompts containing the owner's name. We find that PLMs do leak per-sonal information due to memorization. Howsonal information due to memorization. How ever, since the models are weak at association extracted by attackers is low. We hope this derstand the privacy risk of PLMs and bring

Pre-trained Language Models (PLMs) (Devlin ies also suggest that these large models pose some privacy risks. For instance, an adversary is able to To answer the above question, we first idenrecover training examples containing an individual tify two capacities that may cause privacy leakage person's name, email address, and phone number by querying the model (Carlini et al., 2021). This information, thus the information can be recovered may lead to privacy leakage if the model is trained with a specific prefix, e.g., tokens before the information of the model is trained. mation in the training data; and association, i.e.,

PLMs can associate the personal information with 2019). Even if the data is public, PLMs may change its owner, thus attackers can query the information the intended use, e.g., for information that we share but do not expect to be disseminated.

with the owner's name, e.g., the email address of Tom is ____. If a model can only memorize but not

Carlini et al. (2021, 2022) demonstrate that associate, though the sensitive information may be PLMs memorize a lot of training data, so they are prone to leaking privacy. However, if the memorize to leaked in some randomly generated text as shown in Carlini et al. (2021), attackers cannot effectively rized information cannot be effectively extracted, it extract specific personal information since it is dit is still difficult for the attacker to carry out effective attacks. For instance, Lehman et al. (2021) attempt

As far as we know, this paper is the first to make ¹Code and data are available at https://github.com/
tieffhj/LM_PersonalInfoleak. "Equal contribution.

We focus on studying a specific kind of personal

with which they are associated from a BERT model that is pre-trained over clinical notes. However, they find that with their methods, the model canwhich suggests that PLMs may not be prone to

et al., 2019; Brown et al., 2020; Oiu et al., 2020) Based on existing research, we are not sure have taken a significant leap in a wide range of NLP tasks, attributing to the explosive growth of parameters and training data. However, recent studin: Are Large Pre-Trained Language Models Prone

Huang et al., 2022

Any Form of Leakage

Our Focus

We study PII leakage in the presence of privacy mechanisms such as Differential Privacy or PII Scrubbing

Extraction	Reconstruction	Inference
Black-box Model Access	 Black-box Model Access Masked Training Data 	 Black-box Model Access Masked Training Data Auxiliary Information

Is differential privacy alone sufficient to protect PII?

Security Games for PII Leakage

- $S \sim \{S \in D | \text{EXTRACT}(S) \neq \emptyset\}$
- $C \sim \text{EXTRACT}(S)$
- $\tilde{C} \leftarrow \mathcal{A}(\mathcal{T}, \mathcal{D}, n, \mathcal{O}_{\theta}(\cdot), \text{SCRUB}(\text{SPLIT}(S, C)))$

- $C \sim \text{EXTRACT}(S)$
- $\mathcal{C} \sim \mathcal{E}^m$
- $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$
- $\tilde{C} \leftarrow \mathcal{A}(\mathcal{T}, \mathcal{D}, n, \mathcal{O}_{\theta}(\cdot), \text{SCRUB}(\text{SPLIT}(S, C)), \mathcal{C})$

Setup

Training Dataset

John Doe is a doctor in London

John Doe lives on Sunset Street



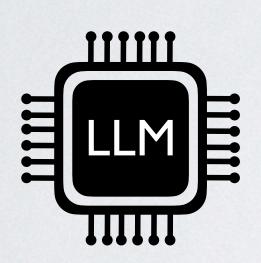
Testing Dataset

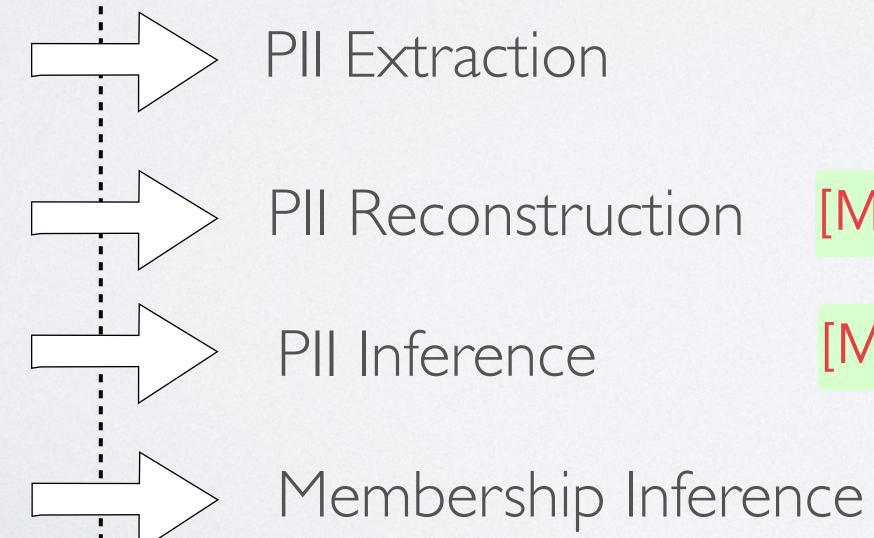
John is a doctor from Sunset Street

John Doe works in London



- Training Training
 2. DP
 Procedure
 3. Scrubbing
 4. DP & Scrubbing
 - No Defense
- Small
- 2. Medium
- Large XL
- Enron
- 2. Yelp-Health
- ECHR





[MASK] lives on Sunset Street



[MASK] lives on Sunset Street



John Doe or Joe Peric

Datasets with many Detectable PII

	Records	Tokens / Record	Unique PII	Records w. PII	Duplicates / PII	Tokens / PII
ECHR	118 161	88.12	16 133	23.75%	4.66	4.00
Enron	138 919	346.10	105 880	81.45%	11.68	3.00
Yelp-Health	78 794	143.92	17 035	54.55%	5.35	2.17

ECHR - European Court for Human Rights

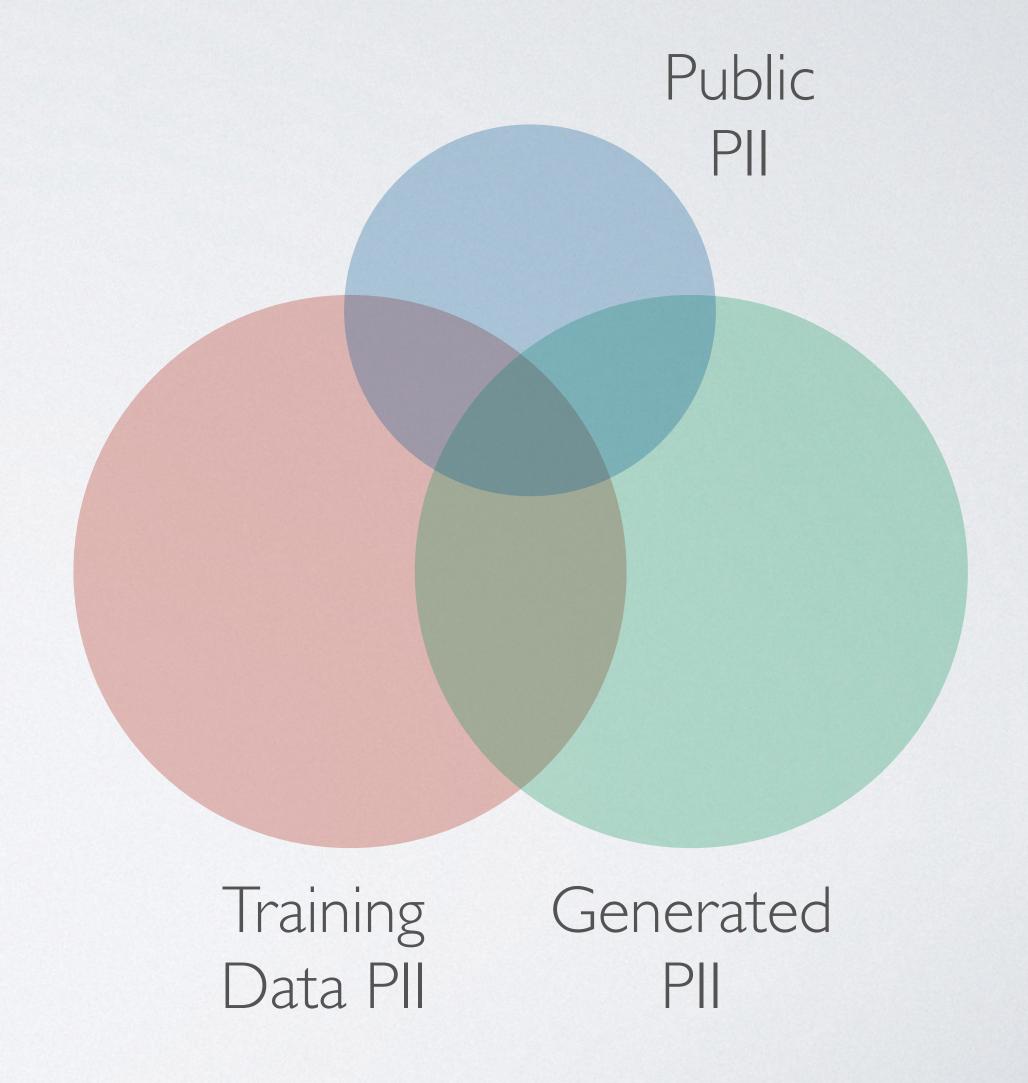
Enron - Corporate e-mails

Yelp-Health - Reviews for healthcare facilities

Extraction Attack

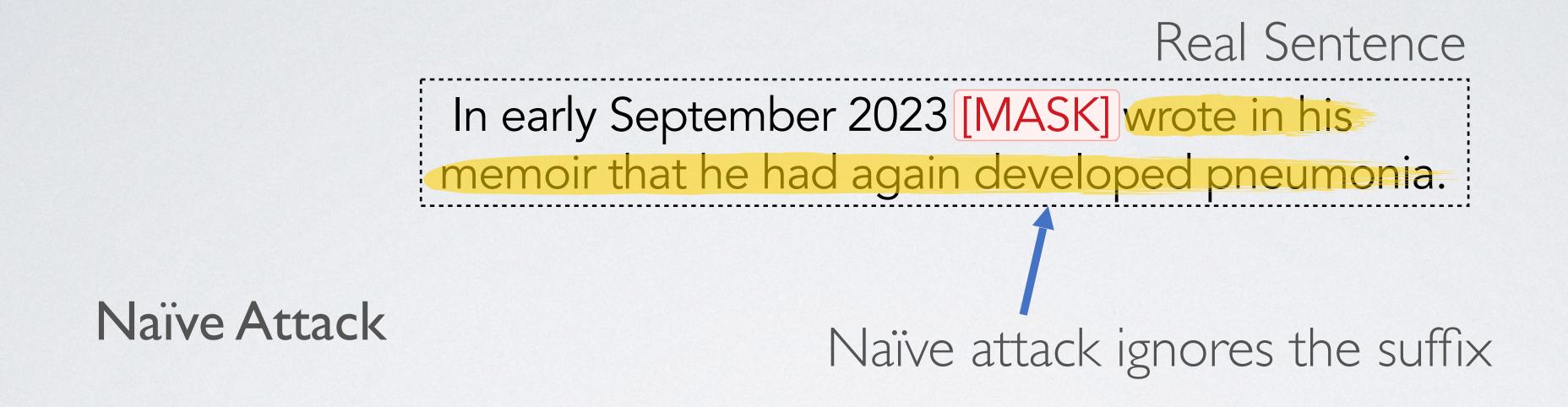
Goal: Extract PII from
Training data with no
auxiliary information

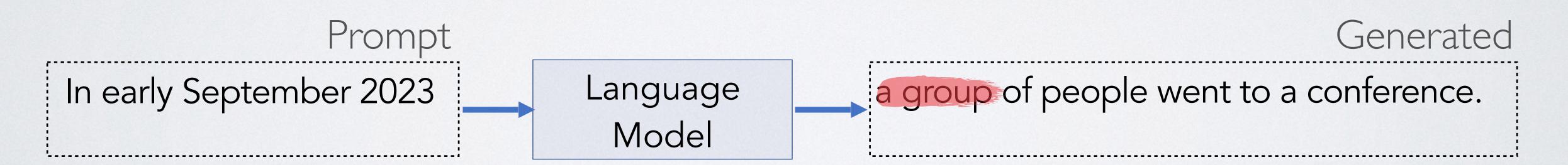
- 1. Generate N sequences with the model
- 2. Tag PII generated by the model
- 3. Calculate Precision & Recall



Reconstruction Attack

Goal: Reconstruct PII given a masked sentence From the training data





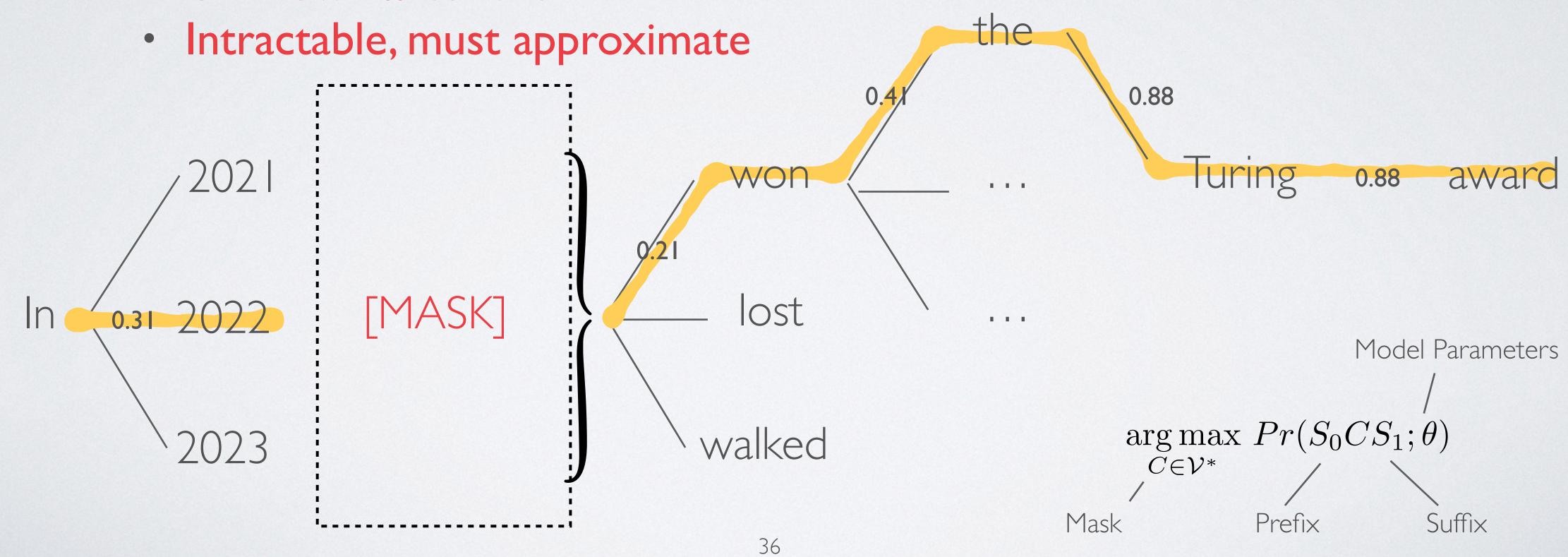
Reconstruction Attack Intuition

Goal: Reconstruct PII given a masked sentence From the training data

Real Sentence

In 2022 [MASK] won the Turing award.

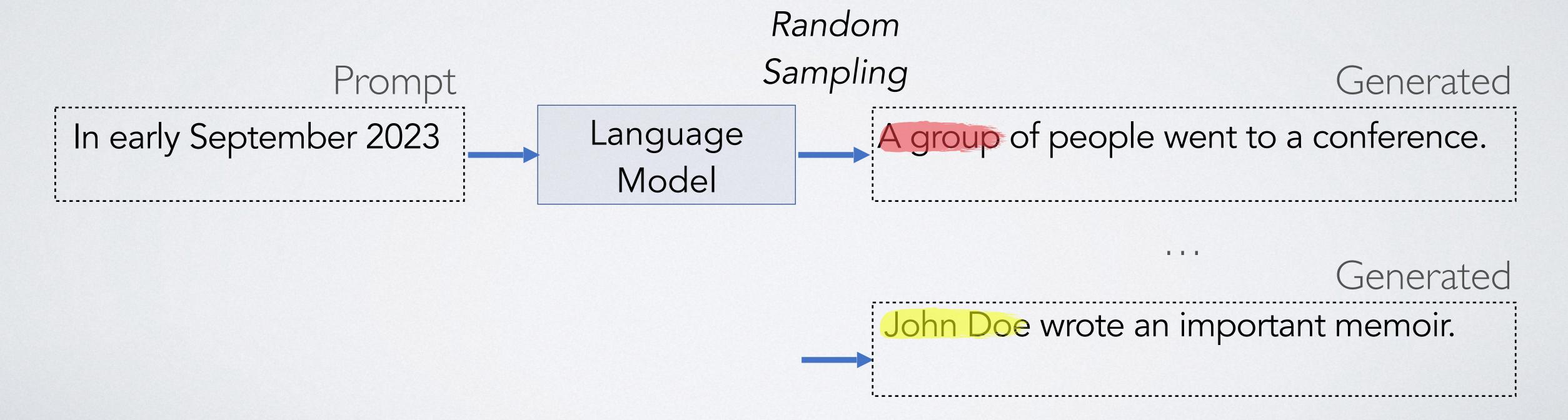
Unknown # tokens



Reconstruction Attack



In early September 2023 [MASK] wrote in his memoir that he had again developed pneumonia.



Reconstruction Attack

Real Sentence

In early September 2023 [MASK] wrote in his memoir that he had again developed pneumonia.

Generated

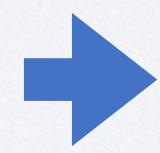
A group of people went to a conference.

. . .

Generated

John Doe wrote an important memoir.

Tag PII



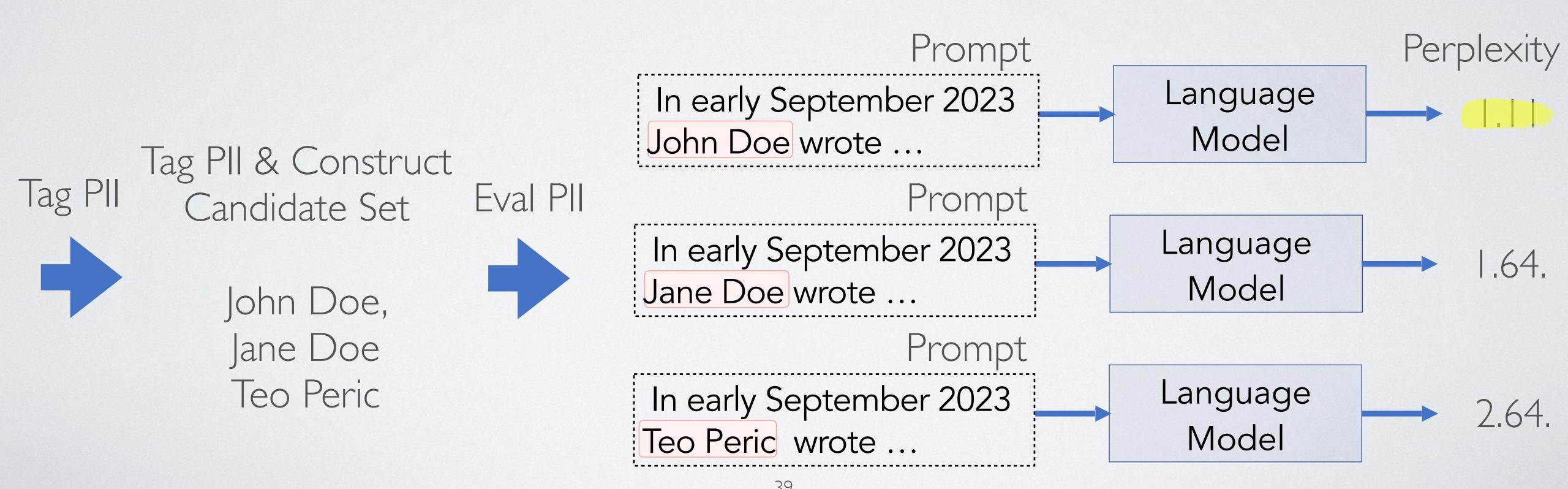
Tag PII & Construct
Candidate Set

John Doe, Jane Doe Teo Peric

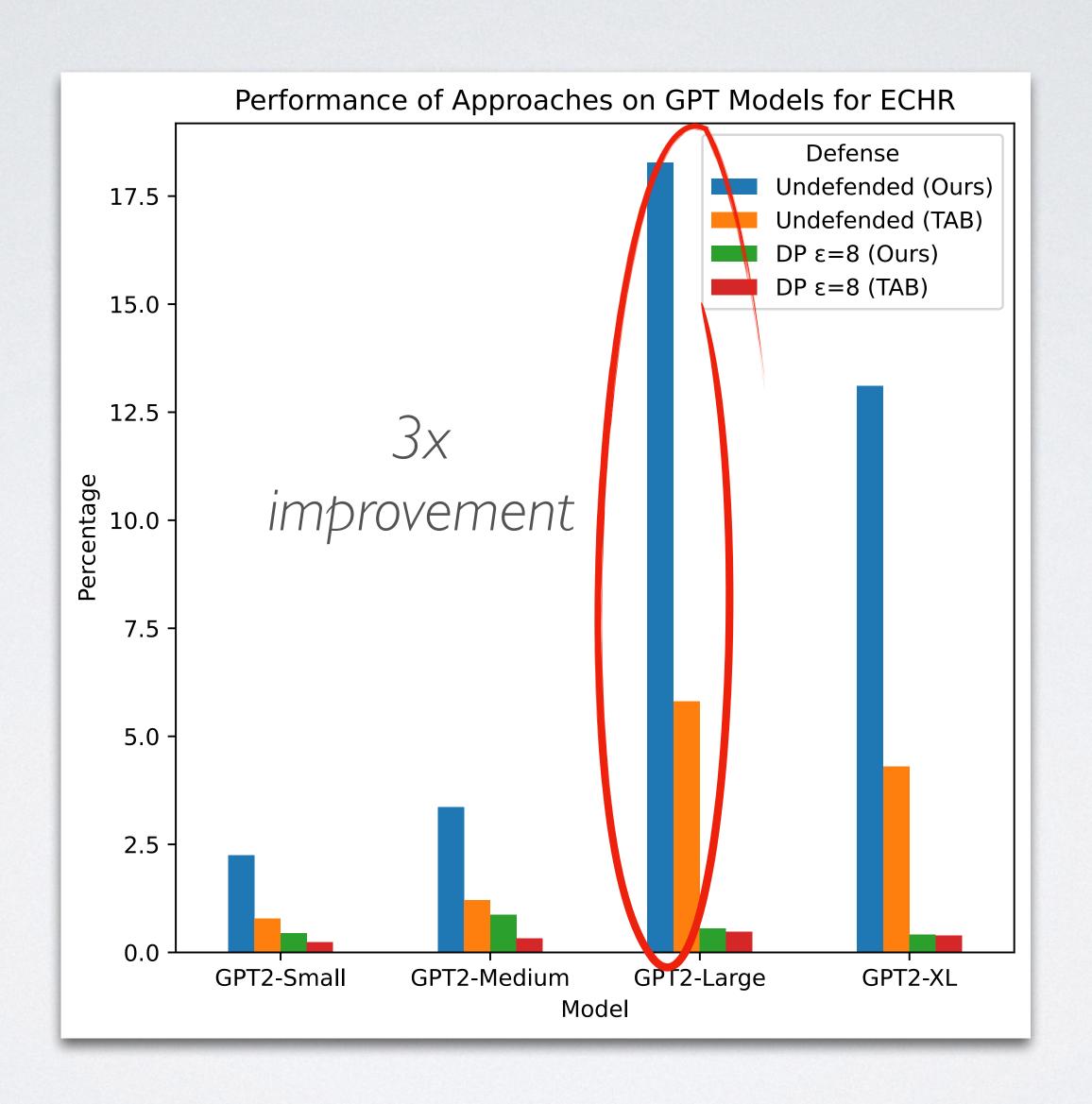
Reconstruction Attack

Real Sentence

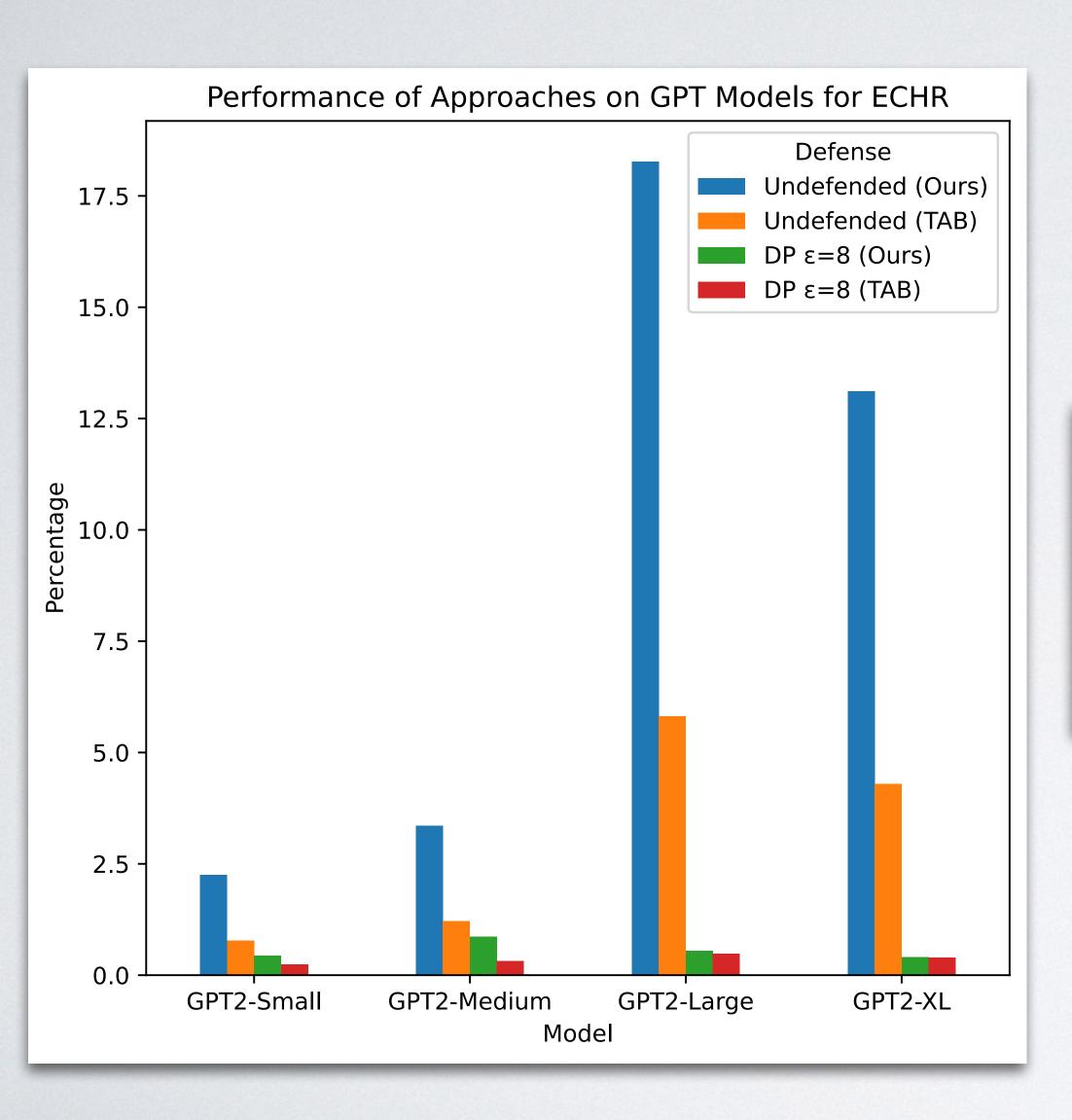
In early September 2023 [MASK] wrote in his memoir that he had again developed pneumonia.



PII Reconstruction



PII Reconstruction



	GPT2-Small		GPT2-Medium		GPT2-Large		GPT2-XL	
	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$
ECHR(TAB)	0.78%	0.24%	1.21%	0.32%	5.81%	0.48%	4.30%	0.39%
ECHR (Ours, $ \mathcal{C} = 64$)	2.25%	0.44%	3.36%	0.87%	18.27%	0.55%	13.11%	0.41%
Enron (TAB)	0.59%	0.04%	0.67%	0.04%	1.75%	0.04%	2.19%	0.19%
Enron (Ours, $ \mathcal{C} = 64$)	6.29%	0.49%	7.26%	0.52%	12.68%	0.55%	15.25%	0.53%
Yelp-Health (TAB)	0.33%	0.24%	0.37%	0.14%	0.65%	0.12%	1.99%	0.12%
Yelp-Health (Ours, $ \mathcal{C} = 64$)	0.42%	0.32%	1.31%	0.32%	1.69%	0.35%	6.40%	0.36%

up to 7x Improvement

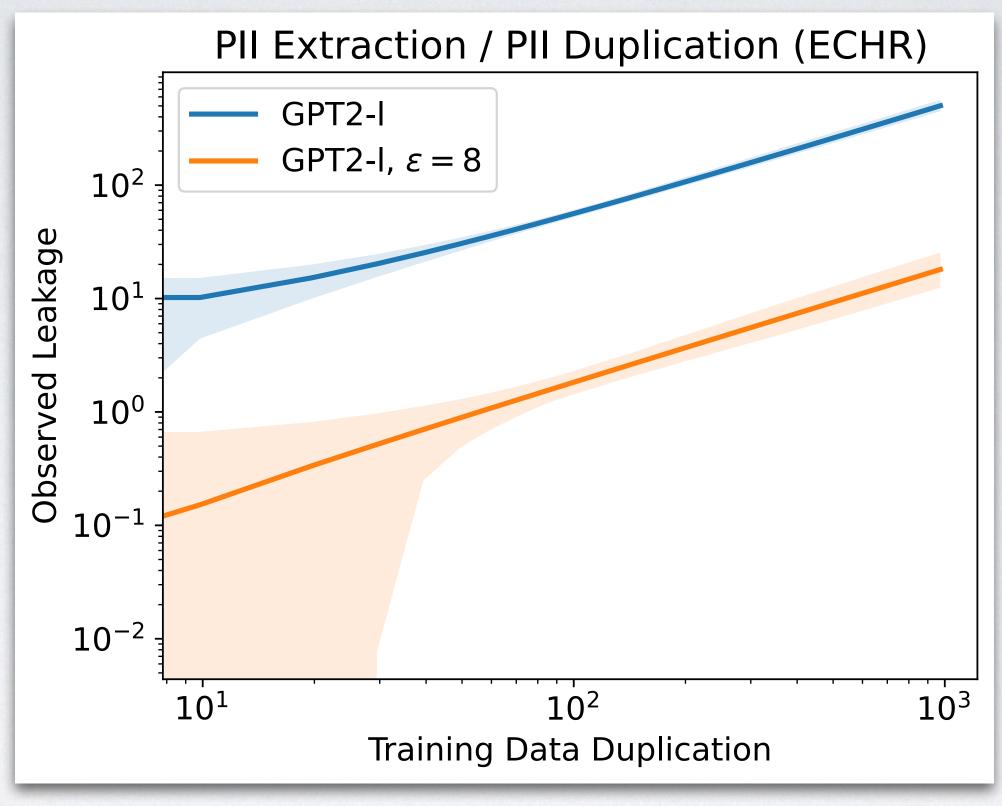
PII Inference

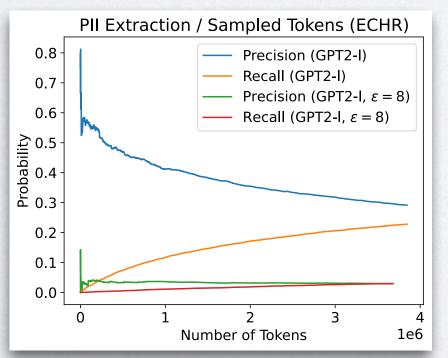
Goal: Infer PII given a masked sentence from the training data a set of PII candidates

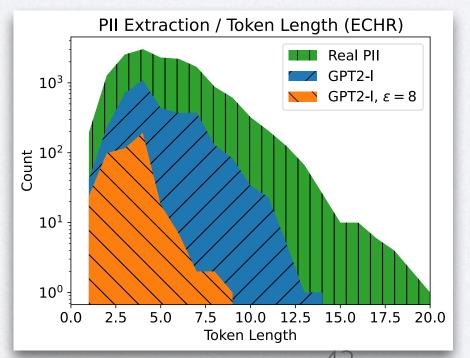
	ECHR		Enron		Yelp-Health	
	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$
$ \mathcal{C} = 100$	70.11%	8.32%	50.50%	3.78%	28.31%	4.29%
$ \mathcal{C} = 500$	51.03%	3.71%	34.14%	1.92%	15.55%	1.86%

Duplicated PII are Extractable more often

(Linear scaling)

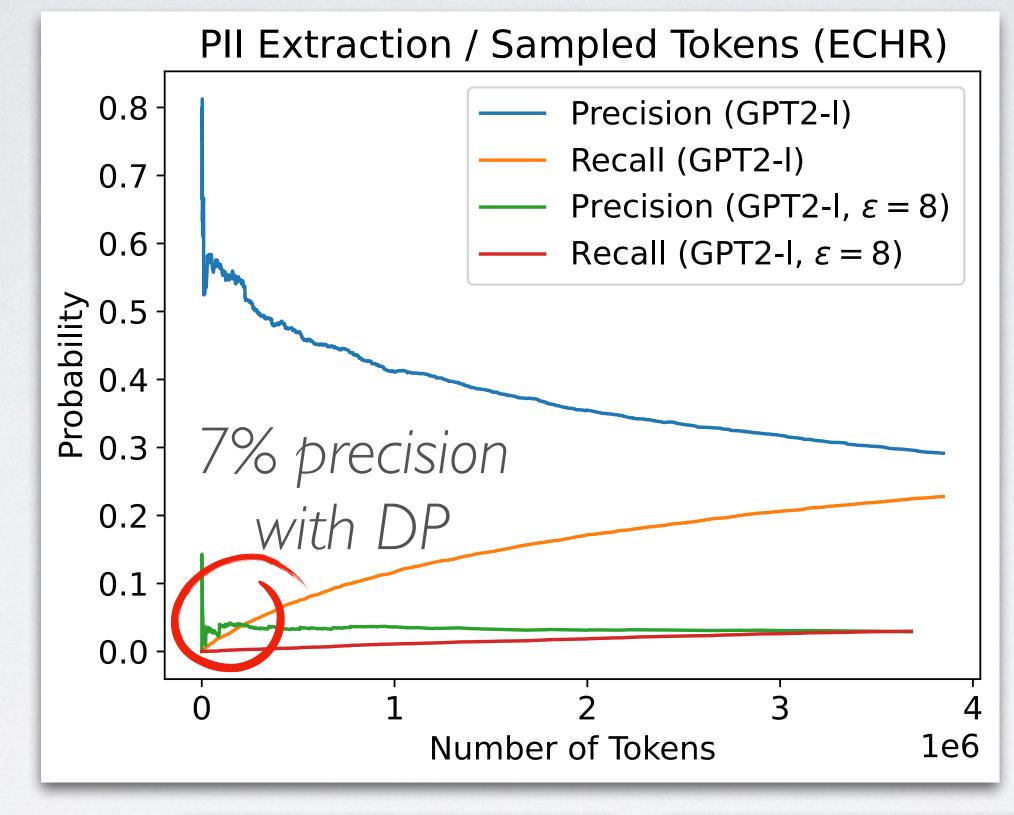


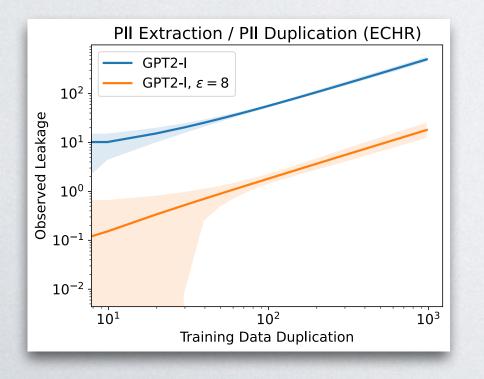


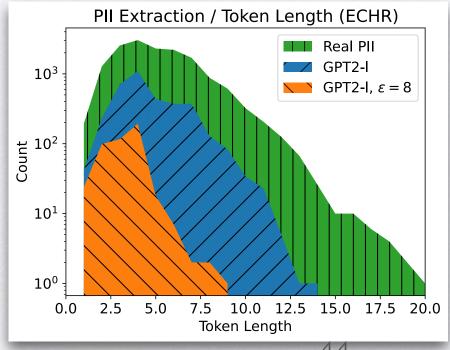


	GPT2-Small		GPT2-N	Medium	GPT2-Large				
	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$			
ECHR									
Prec	24.91%	2.90%	28.05%	3.02%	29.56%	2.92%			
Recall	9.44%	2.98%	12.97%	3.21%	22.96%	2.98%			
	Enron								
Prec	33.86 %	9.37%	27.06%	12.05%	35.36%	11.57%			
Recall	6.26%	2.29%	6.56%	2.07%	7.23%	2.31%			
	Yelp-Health								
Prec	13.86%	8.31%	14.87%	6.32%	14.28%	7.67%			
Recall	11.31%	5.02%	11.23%	5.22%	13.63%	6.51%			

High-precision/ Low-recall attacks

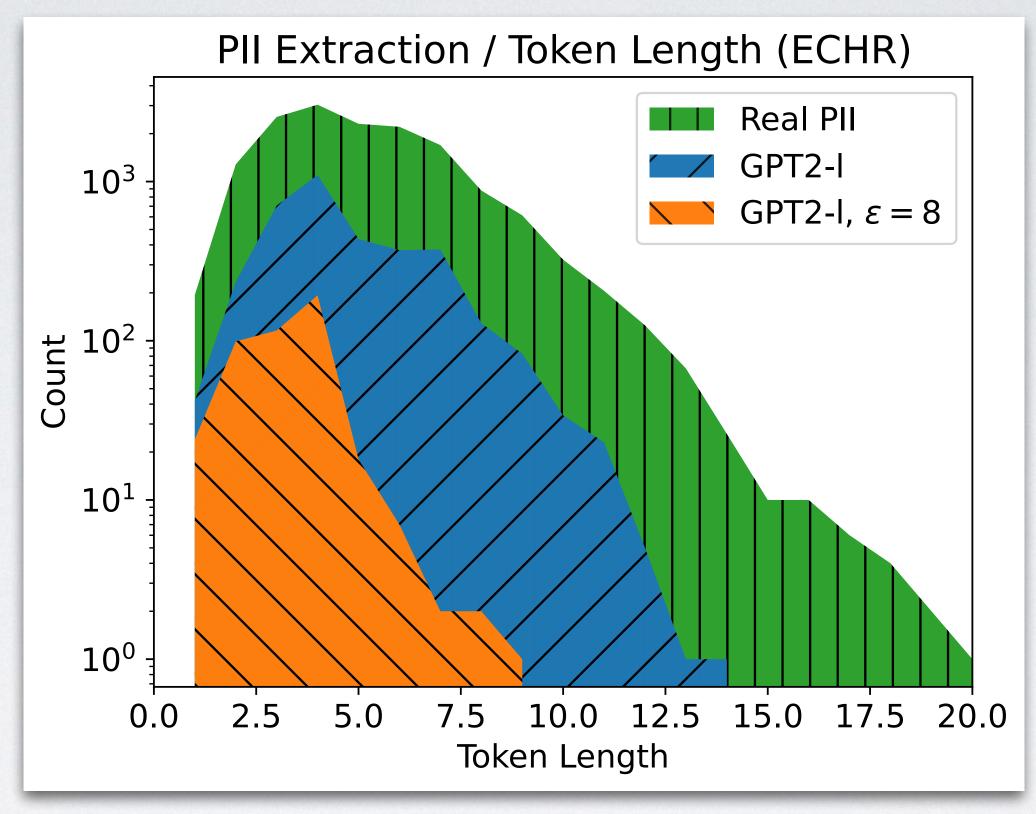


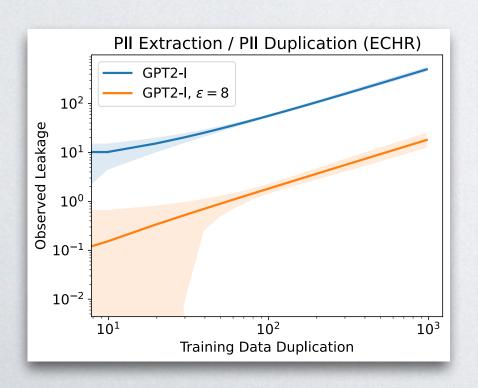


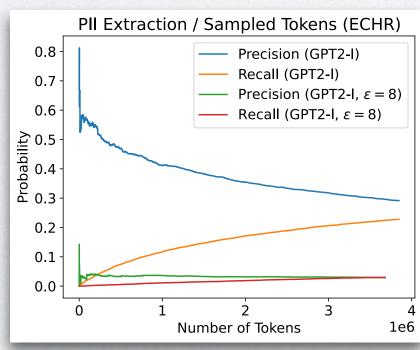


	GPT2-Small		GPT2-N	Medium	GPT2-Large					
·	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$				
	ECHR									
Prec	24.91%	2.90%	28.05%	3.02%	29.56%	2.92%				
Recall	9.44%	2.98%	12.97%	3.21%	22.96%	2.98%				
	Enron									
Prec	33.86 %	9.37%	27.06%	12.05%	35.36%	11.57%				
Recall	6.26%	2.29%	6.56%	2.07%	7.23%	2.31%				
	Yelp-Health									
Prec	13.86%	8.31%	14.87%	6.32%	14.28%	7.67%				
Recall	11.31%	5.02%	11.23%	5.22%	13.63%	6.51%				

PII with many tokens are protected in DP models



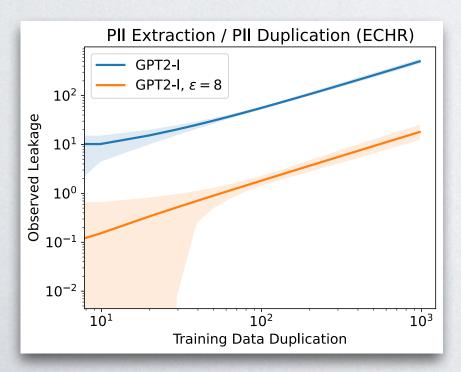


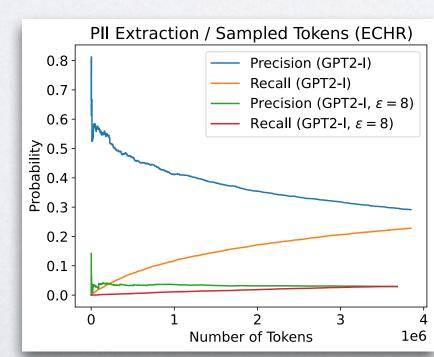


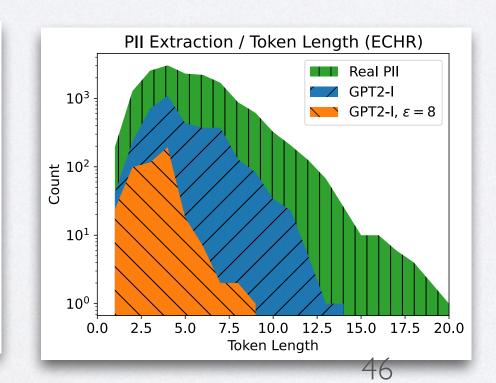
	GPT2-Small		GPT2-N	Medium	GPT2-Large				
	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$			
ECHR									
Prec	24.91%	2.90%	28.05%	3.02%	29.56%	2.92%			
Recall	9.44%	2.98%	12.97%	3.21%	22.96%	2.98%			
	Enron								
Prec	33.86 %	9.37%	27.06%	12.05%	35.36%	11.57%			
Recall	6.26%	2.29%	6.56%	2.07%	7.23%	2.31%			
	Yelp-Health								
Prec	13.86%	8.31%	14.87%	6.32%	14.28%	7.67%			
Recall	11.31%	5.02%	11.23%	5.22%	13.63%	6.51%			

Higher recall in larger models

	GPT2-Small		GPT2-I	Medium	GPT2-Large				
	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$			
	ECHR								
Prec	24.91%	2.90%	28.05%	3.02%	29.56%	2.92%			
Recall	9.44%	2.98%	12.97%	3.21%	22.96%	2.98%			
	Enron								
Prec	33.86 %	9.37%	27.06%	12.05%	35.36%	11.57%			
Recall	6.26%	2.29%	6.56%	2.07%	7.23%	2.31%			
	Yelp-Health								
Prec	13.86%	8.31%	14.87%	6.32%	14.28%	7.67%			
Recall	11.31%	5.02%	11.23%	5.22%	13.63%	6.51%			

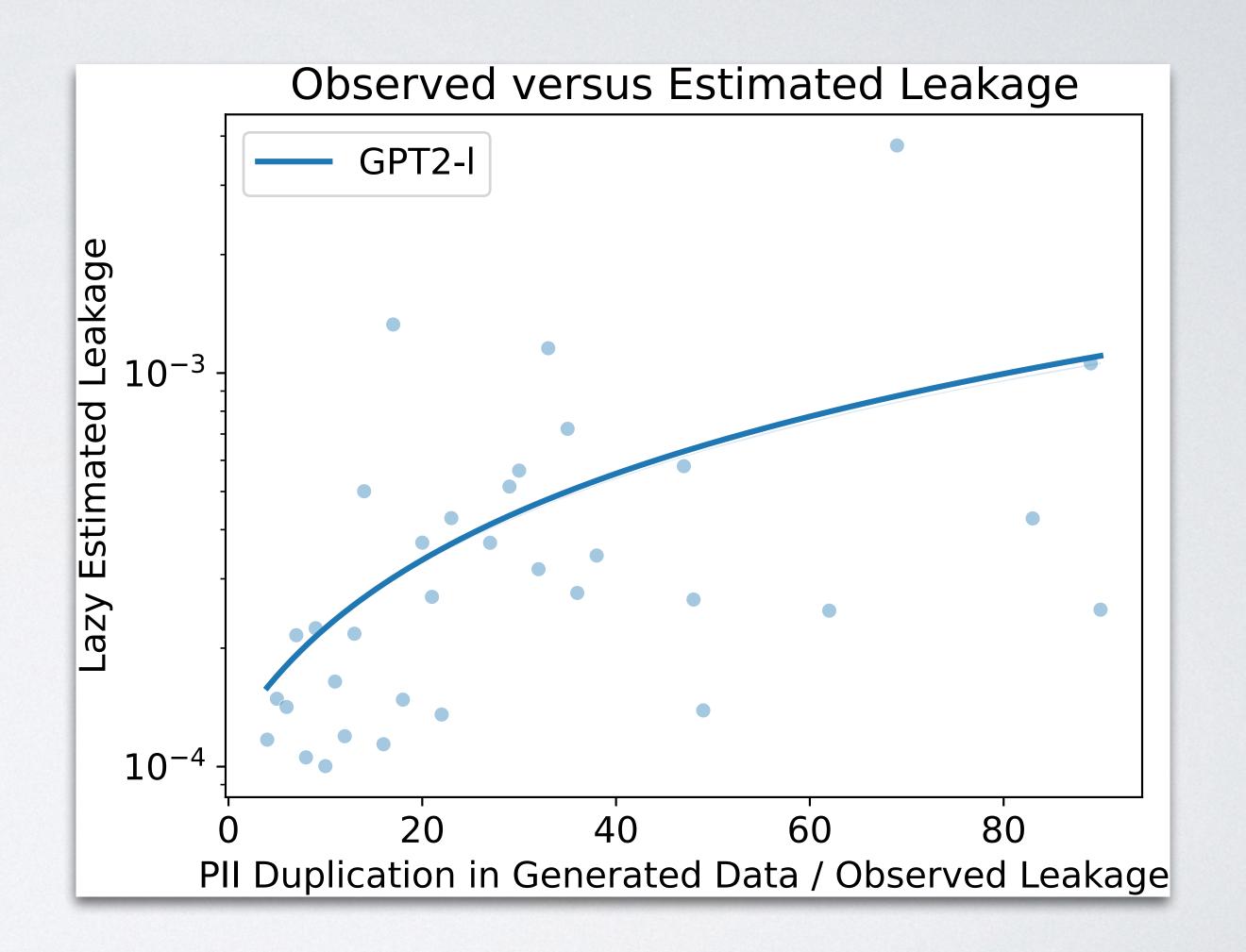






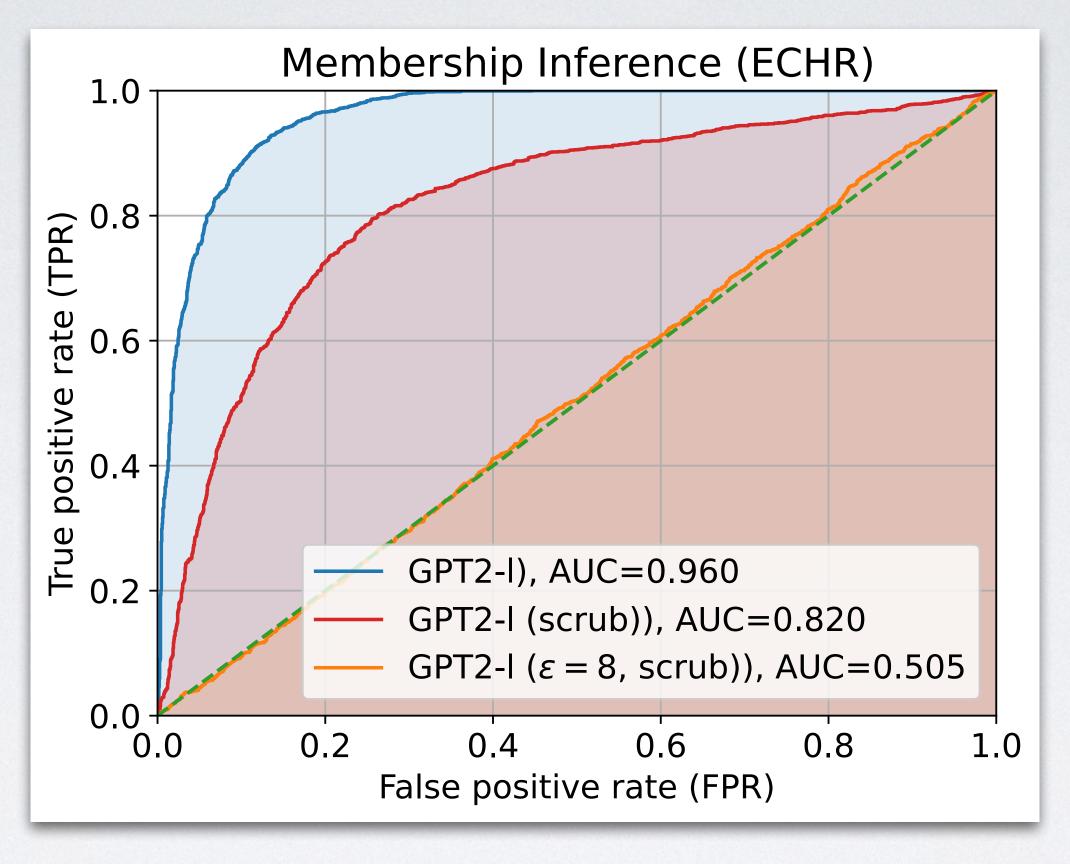
Estimating Extractability

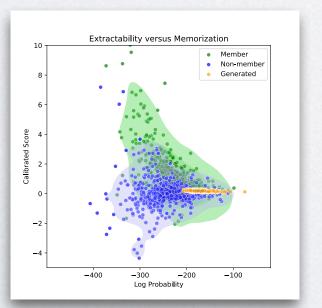
Once upon a time, there existed a tale of two medical students. In the year 2022, they resided at Sunset Street while pursuing their medical education. Alongside his friend, he worked at the LHS Hospital located in the bustling heart of downtown London. Before donning their white coats, both **TeberPoic** and ...

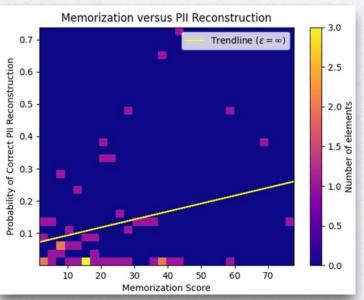


Membership Inference

Scrubbing does not prevent MI

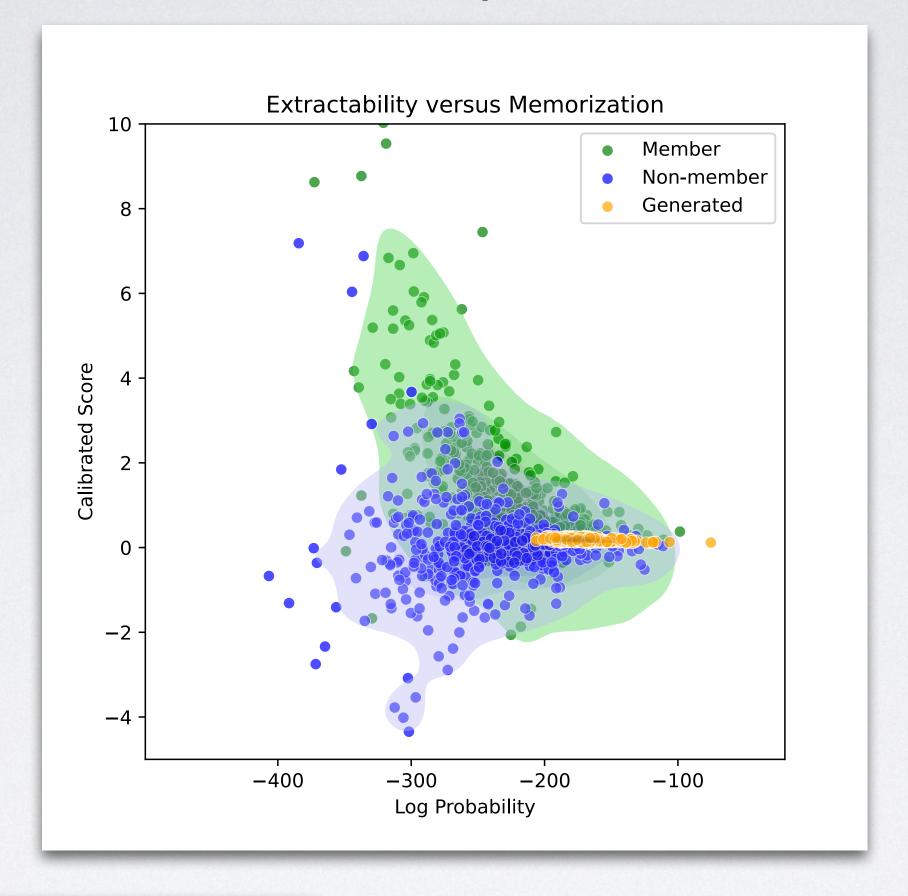


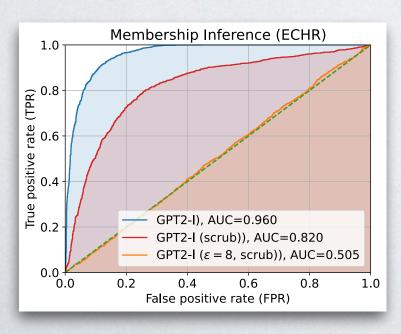


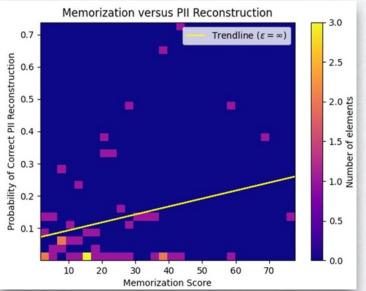


Membership Inference

Randomly generated sequences likely do not contain MI signal

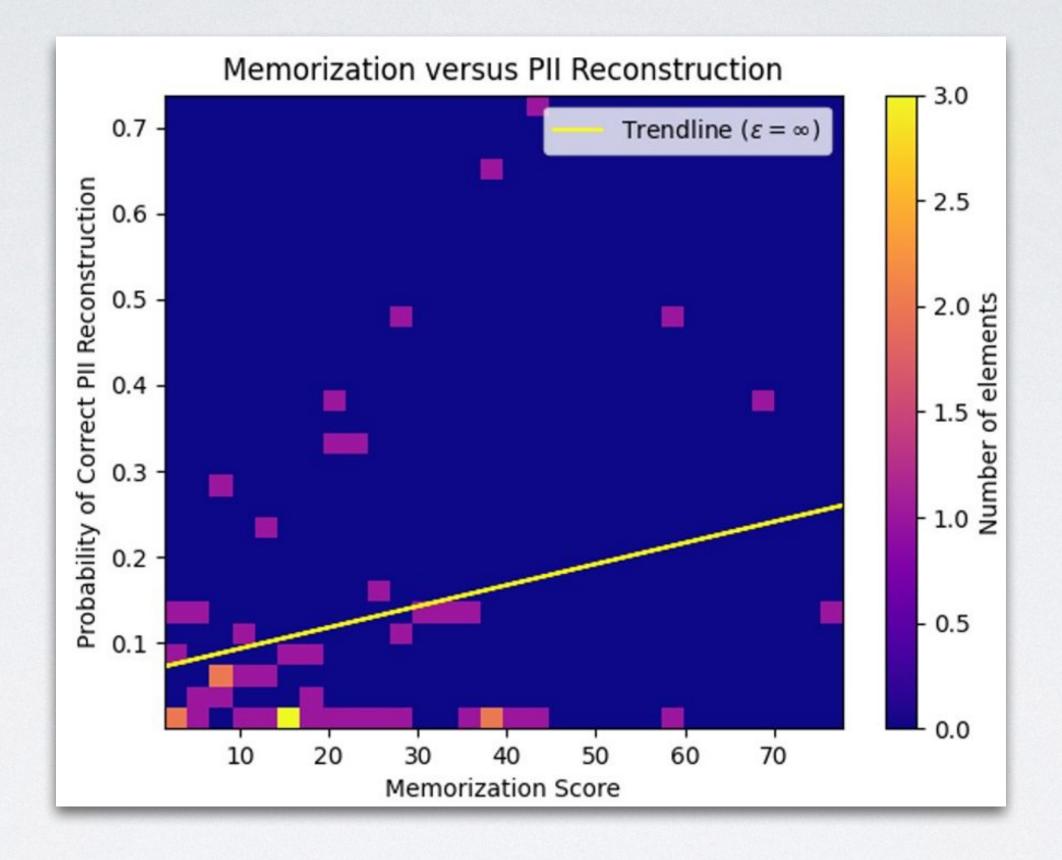


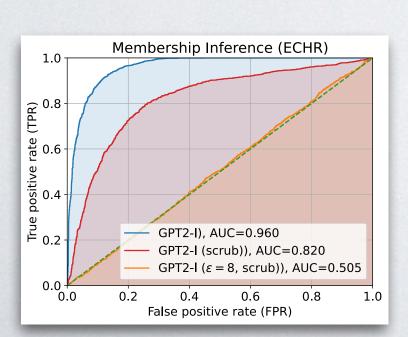


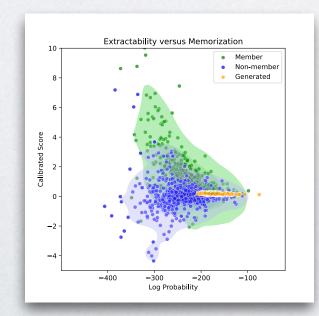


Membership Inference & PII Reconstruction

MI correlates with PII reconstruction







Summary of Results

Metric	Undefended	DP	Scrub	DP + Scrub
Test Perplexity	9	140	16	16
Extract Precision	30%	3%	0%	0%
Extract Recall	23%	3%	0%	0%
Reconstruction Acc.	18%	1%	0%	0%
Inference Acc. ($ C = 100$)	70%	8%	1%	1%
MIAUC	0.96	0.5	0.82	0.5

Limitations

- (General Applicability) We focus on fine-tuned GPT-2 Language Models (0.12b to 1.7b parameters)
- (Syntactic Similarity) We consider only verbatim leakage (i.e., "John Doe" and "J. Doe" are different)
- (PII Association) Our extraction attacks study leakage in isolation (single PII, no association between PII)
- (Need for better Benchmarks) Our study is limited by the quality of the NER tools used; Evaluating scrubbing methods requires large, annotated datasets

Outlook

We take a number of steps to reduce the risk that our models are used in a way that could violate a person's privacy rights. These include fine-tuning models to reject these types of requests, removing personal information from the training dataset where feasible, creating automated model evaluations, monitoring and responding to user attempts to generate this type of information, and restricting this type of use in our terms and policies. Our efforts to expand context length and improve embedding models for retrieval may help further limit privacy risks moving forward by tying task performance more to the information a user brings to the model. We continue to research, develop, and enhance technical and process mitigations in this area.

GPT-4 Technical Report, 2023 [8]

Scrubbing?	1) D	Synthetic data?
Fake PII?	1) Data sanitation2) Alignment	Lower epsilon?
Stronger attacks / audits?	3) Model evaluation	Know your user?
Unlearning?	4) Safety filters	Smaller models?
Regularization?	53	Red teaming?

Check out our Paper for more Information

Analyzing Leakage of Personally Identifiable Information in Language Models

Nils Lukas*§, Ahmed Salem[†], Robert Sim[†], Shruti Tople[†], Lukas Wutschitz[†] and Santiago Zanella-Béguelin[†] *University of Waterloo, †Microsoft

nlukas@uwaterloo.ca, {t-salemahmed, rsim, shruti.tople, lukas.wutschitz, santiago}@microsoft.com

Abstract-Language Models (LMs) have been shown to leak information about training data through sentence-level membership inference and reconstruction attacks. Understanding the risk of LMs leaking Personally Identifiable Information (PII) has received less attention, which can be attributed to the false assumption that dataset curation techniques such as scrubbing are sufficient to prevent PII leakage. Scrubbing techniques reduce but do not prevent the risk of PII leakage: in practice scrubbing is imperfect and must balance the trade-off between minimizing disclosure and preserving the utility of the dataset. On the other hand, it is unclear to which extent algorithmic defenses such as differential privacy, designed to guarantee sentenceor user-level privacy, prevent PII disclosure. In this work, we introduce rigorous game-based definitions for three types of PII leakage via black-box extraction, inference, and reconstruction attacks with only API access to an LM. We empirically evaluate the attacks against GPT-2 models fine-tuned with and without defenses in three domains: case law, health care, and e-mails. Our main contributions are (i) novel attacks that can extract up to 10× more PII sequences than existing attacks, (ii) showing that sentence-level differential privacy reduces the risk of PII disclosure but still leaks about 3% of PII sequences, and (iii) a subtle connection between record-level membership inference and PII reconstruction. Code to reproduce all experiments in the paper is available at https://github.com/microsoft/analysing_pii_leakage.

I. INTRODUCTION

Language Models (LMs) are fundamental to many natural language processing tasks [22, 49]. State-of-the-art LMs scale to trillions of parameters [19] and are pre-trained on large text corpora (e.g., 700GB [53]). Pre-trained LMs are adapted to downstream tasks by fine-tuning on domain-specific datasets such as human dialogs [7] or clinical health data [62] which

LM. Extracting any PII by itself, such as a personal address, may contain private information.

Memorization is a privacy concern in LMs [9]. The threat is that an attacker learns by whom the training data was for example, "In May 2022, [MASK] had chemother provided, known as membership inference [30, 45, 46, 58] and about whom it contains information, known as data extraction [9, 11, 29, 59, 69]. These two categories can be attacks proposed in this work. disjoint but associations in the latter can be used to infer information about the former. For LMs, data extraction is a significant threat in practice since attackers with black-box API access can extract at least 1% of the training data [11].

Existing work focuses on finding a lower bound on any kind of memorization but does not differentiate public and private

§Part of this work was done during an internship at Microsoft Research.

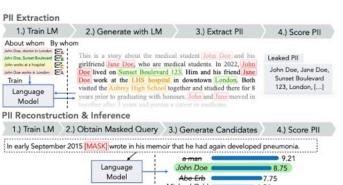


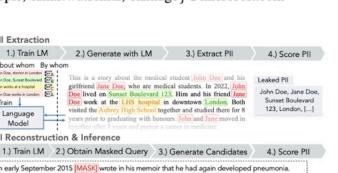
Fig. 1: An illustration of PII extraction, reconstruction and

leaked information. For example, leaking highly duplicated common phrases is not a privacy violation according to the GDPR [17] as opposed to leaking Personally Identifiable Information (PII). In practice, any LM trained on real, sensitive data has to protect PII, but memorization of PII is not well understood. We believe that a comprehensive study on the risk of PII memorization in LMs is missing.

Consider a service provider who wants to deploy a nextword prediction LM for composing e-mails, such as Google's Smart Compose [13]. Their goal is to train an LM with high utility that does not leak PII and make it available as a blackbox API. The threat is an attacker who learns PII, such as names, addresses or other sensitive information through the can already pose a privacy threat. This threat is elevated when an attacker can associate a piece of PII to a context,

LHS". As a part of this paper, attacks on LMs in practice. Figu

curation technique that removes PII from text, relying on Named Entity Recognition (NER) [35] to tag PII. Modern NER is based on the Transformer architecture [63] and has mixed recall of 97% (



if and 16.3% of mentioned organization names from an in f models (< | Dollar | our attack still approves the larger | Dollar | our attack still approves the larger | Dollar | our attack still approves the larger | our attack still approves | our attack still approve

phone number in the DP model. However, we observe leakage compared to TAB becomes more evident. E-mails in the Enron of e-mail addresses (consisting of equally many tokens), that dataset typically mention the receiver of the e-mail at the

Security pi games soil of the list is lead to be seen to be soil of the list in the lead to be seen to be seen

sample's suffix provides a strong signal to reconstruct PII. On

ECHR, our attack improves the baseline by at least $2.5\times$, on

Enron we observe an improvement of at least $7.5 \times$ and on

We sample 64 candidates and decode from the model using limits our attack's success rate. We believe a method that top-k sampling with k=40. We observe that our reconsamples candidates by incorporating information from the

No DP $\varepsilon = 8$ No DP $\varepsilon = 8$ No DP $\varepsilon = 8$

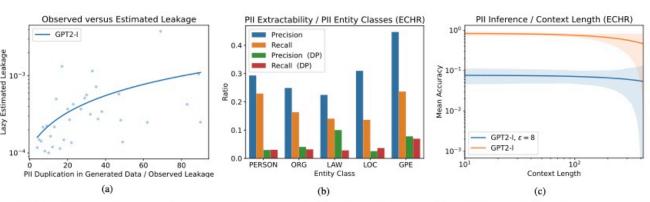


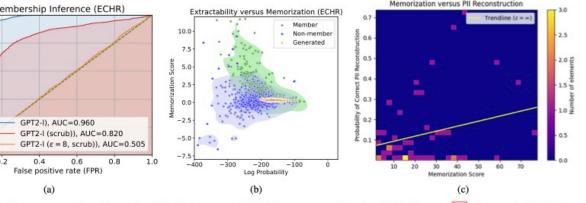
Fig. 6: Fig. 6a shows the correlation between the observed and estimated leakage. Fig. 6b shows the precision and recall for other entity classes on the ECHR dataset. Fig. 6c shows the mean inference accuracy relative to the context length, which is the length of the combined prefix and suffix for a masked query.

	GPT2-Small		GPT2-Medium		GPT2-Large		GPT2-XL	
	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$	No DP	$\varepsilon = 8$
ECHR(TAB)	0.78%	0.24%	1.21%	0.32%	5.81%	0.48%	4.30%	0.39%
ECHR (Ours, $ C = 64$)	2.25%	0.44%	3.36%	0.87%	18.27%	0.55%	13.11%	0.41%
Enron (TAB)	0.59%	0.04%	0.67%	0.04%	1.75%	0.04%	2.19%	0.19%
Enron (Ours, $ \mathcal{C} = 64$)	6.29%	0.49%	7.26%	0.52%	12.68%	0.55%	15.25%	0.53%
Yelp-Health (TAB)	0.33%	0.24%	0.37%	0.14%	0.65%	0.12%	1.99%	0.12%
Yelp-Health (Ours, $ \mathcal{C} = 64$)	0.42%	0.32%	1.31%	0.32%	1.69%	0.35%	6.40%	0.36%

TABLE IV: Results of PII reconstruction attacks on the entity class "person". Bold numbers represent the best attack per dataset and LM. We compare our results with the TAB attack [28] on three datasets.

further research motivated by our findings: how to best combine DP training and scrubbing, optimizing attacks for other

PILExtraction, Reconstruction and Inference Attacks



but orthogonal problem.

Fig. 7: Connecting sentence-level membership inference with PII reconstruction in GPT-2-Large. 7a shows the ROC curve against our fine-tuned model using a shadow model attack on ECHR. 7b shows that the memorization score of generated sequences is nearly zero and 7c shows that the memorization score correlates with the probability of correct PII reconstruction.

	Undefended	DP	Scrub	DP + Scrub
Test Perplexity	14 / 9	14	16	16
Extract Precision	30%	3%	0%	0%
Extract Recall	23%	3%	0%	0%
Reconstruction Acc.	18%	1%	0%	0%
Inference Acc. ($ C = 100$)	70%	8%	1%	1%
MI AUC	0.96	0.5	0.82	0.5

TABLE VI: Our results on ECHR for GPT-2-Large summarize the privacy-utility trade-off. We show the undefended model's perplexity with/without masking generated PII. The undefended model has the lowest perplexity but the highest leakage. DP with $\epsilon = 8$ mitigates MI and (partially) PII leakage. Scrubbing only prevents PII leakage. DP with scrubbing mitigates all the privacy attacks but suffers from utility

• DP does not completely eliminate leakage from PII inference and PII extraction. We demonstrate that an 100 candidates) in a practical setting.

the LM's utility, motivating the search for defenses with that explore associations among them and reveal additional better empirical privacy/utility trade-offs.

V. DISCUSSION AND LIMITATIONS

alent to a naive combination of DP training and scrubbing.

adapted to account for both syntactic and semantic similarity. For example, "Mr. John Doe" and "J. Doe" could be inferred to be the same person. Similarly, PII reconstruction and PII inference attacks can employ contexts with similar meaning to improve attack results Advanced Attacks. We consider leakage of PII sequences attacker can infer PII with up to 10% accuracy (given from the training dataset in isolation, irrespective of the context where it appears and other extracted PII. Extracted We find that DP and (aggressive) PII scrubbing limit PII sequences can be further leveraged in advanced attacks

evaluating leakage of sensitive sequences of tokens categorized

as PII. That said, we bring attention to the point that our

methodology is generally applicable to any notion of sensitive

input. As long as one has an effective method to correctly iden-

tify inputs deemed sensitive, our methodology can be adapted

to measure the protection offered by existing ML pipelines

in mitigating the leakage of any sensitive information. In

practice, it is often hard to draw a clear boundary around

what constitutes sensitive information, which is an important

Syntactic and Semantic Similarity. We consider verbatim

matches of PII tokens as leakage, however, our methods can be

linkability attacks. Utility-preserving Scrubbing. Our empirical evaluation demonstrates that differential privacy is partially effective in mitigating leakage of PII. Based on this observation, existing methodology to scrubbing techniques can be adapted to take into consideration tential extensions to the partial protection offered by DP and heuristically scrub only PII that remains unprotected (e.g. because it occurs many times). Such a DP-informed scrubbing would allow for improving model utility while maintaing a privacy level equiv-

private information about the training dataset, thereby enabling

Comparison to Masked Language Models. Pior work has explored PII reconstruction in the clinical health sets on defining ting [37, 61] with masked language models (MLMs) based on formulas for the BERT architecture [14]. MLMs are trained to reconstruct

*Part of this work was done during an internship at Microsoft Research. *To cite this work, please refer to the full publication (A) in IEEE Security de Privacy (S&P) 2023. *To cite this work, please refer to the full publication (A) in IEEE Security de Privacy (S&P) 2023. *To cite this work, please refer to the full publication (A) in IEEE Security de Privacy (S&P) 2023. *To cite this work, please refer to the full publication (A) in IEEE Security de Privacy (S&P) 2023. *To cite this work, please refer to the full publication (A) in IEEE Security de Privacy (S&P) 2023. *To cite this work, please refer to the full publication (A) in IEEE Security de Privacy (S&P) 2023. *To cite this work, please refer to the full publication (A) in IEEE Security de Privacy (S&P) 2023. *To cite this work, please refer to the full publication (A) in IEEE Security in IEEE S

Connection between Membership Inference and PII Reconstruction

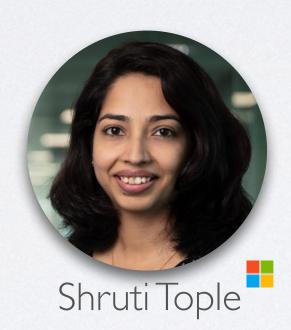
Analyzing Leakage of Personally Identifiable Information in Language Models

Source code: https://github.com/microsoft/analysing-pii-leakage





















GitHub - Source Code



Full Paper

Sources

- [1] https://www.bloomberg.com/news/articles/2023-02-24/citigroup-goldman-sachs-join-chatgpt-crackdown-fn-reports, accessed June 14th
- [2] https://www.businessinsider.in/retail/news/leaked-walmart-memo-warns-employees-not-to-share-any-information-about-walmarts-business-with-chatgpt-or-other-ai-bots/articleshow/98315181.cms, accessed June 14th
 - [3] https://www.bbc.com/news/technology-65 | 39406, accessed June | 4th
 - [5] https://www.bloomberg.com/news/articles/2023-05-02/samsung-bans-chatgpt-and-other-generative-ai-use-by-staff-after-leak, accessed June 14th
 - [6] https://help.openai.com/en/articles/6783457-what-is-chatgpt, accessed June 14th
 - [7] https://bard.google.com/faq?hl=en, accessed June 14th
 - [8] OpenAl, "GPT-4 Technical Report", arXiv preprint arXiv:2303.08774 (2023)
 - [9] https://www.techspot.com/news/56127-10000-aws-secret-access-keys-carelessly-left-in-code-uploaded-to-github.html, accessed June 14th
 - [10] https://analyticsdrift.com/github-copilot-ai-is-leaking-functional-api-keys/, accessed June 14th
 - [11] https://www.bleepingcomputer.com/news/security/github-copilot-update-stops-ai-model-from-revealing-secrets/, accessed June 14th
- [12] Liu, Haokun, et al. "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning." Advances in Neural Information Processing Systems 35 (2022): 1950-1965.

Homepage



https://nilslukas.github.io