

# Analyzing Leakage of Personally Identifiable Information in Language Models

Presented by Peter Li

# Problem Statement

- Models leak Personal Identifiable Information
- Scrubbing techniques reduce but **do not prevent** the risk of PII leakage
- **Unclear** to which extent algorithmic defenses such as differential privacy prevent PII disclosure



# Background

## Scrubbing

A murder has been committed by John D. and [MASK] in a bar close to the Rhine.

## Differential Privacy

$$\Pr(\mathcal{M}(D) \in \mathcal{O}) \leq e^\epsilon \Pr(\mathcal{M}(D') \in \mathcal{O}) + \delta$$

# Background: NER Recall on PII

PHI Type	# Predicted Instances	NER Recall	NER Precision
<i>Health Care Unit</i>	19,659,127	80%	87%
<i>Partial Date</i>	19,374,711	83%	94%
<i>Last Name</i>	14,332,309	97%	96%
<i>First Name</i>	12,525,688	97%	98%
<i>Full Date</i>	10,459,935	55%	77%
<i>Location</i>	3,158,031	89%	85%
<i>Age</i>	2,064,111	35%	47%
<i>Organisation</i>	1,078,115	36%	71%
<i>Phone Number</i>	1,262,313	40%	63%

Table 1: The PHI types in order of frequency as classified by the de-identification system. The per-class recall and precision for the NER model are also displayed and were calculated on the test data from Dalianis and Velupillai (2010). In total, 83,914,340 sensitive entities are found in 49,715,558 sentences.

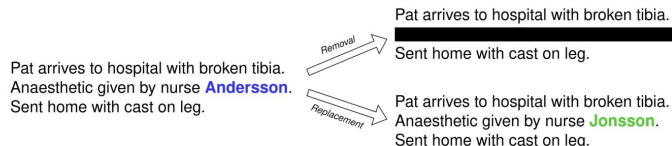


Figure 1: This hypothetical example illustrates the two approaches taken to de-identify the data. One approach *replaces* the sensitive data with realistic surrogates and is used to train the model *KB-BERT + Pseudo*. The other approach instead *removes* the entire sentence from the dataset and this filtered dataset is used to train the model *KB-BERT + Filtered*.

# Threat Model

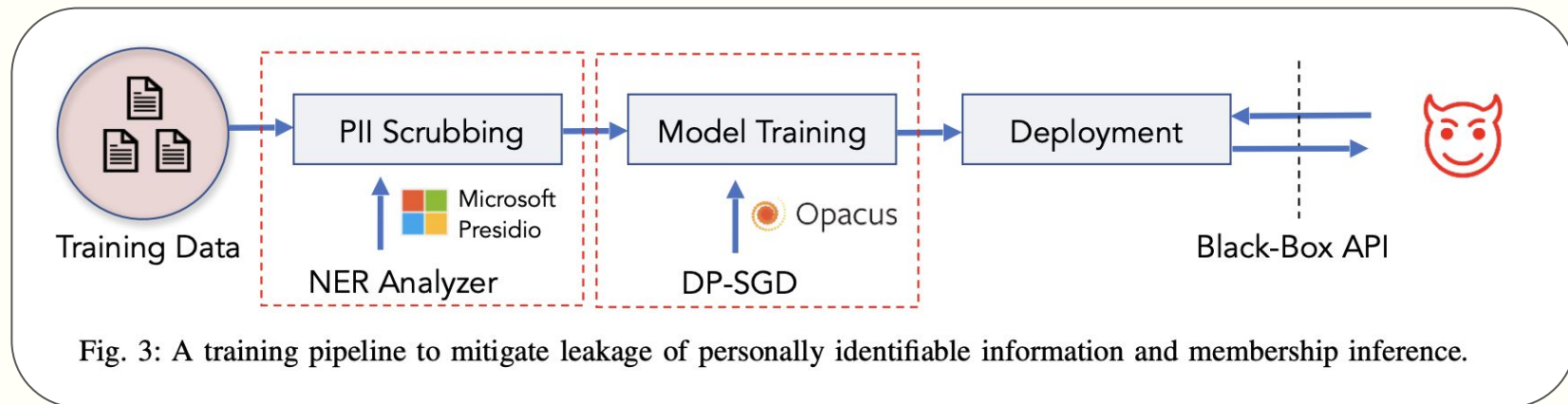


Fig. 3: A training pipeline to mitigate leakage of personally identifiable information and membership inference.

# PII Extraction & PII Reconstruction & Inference

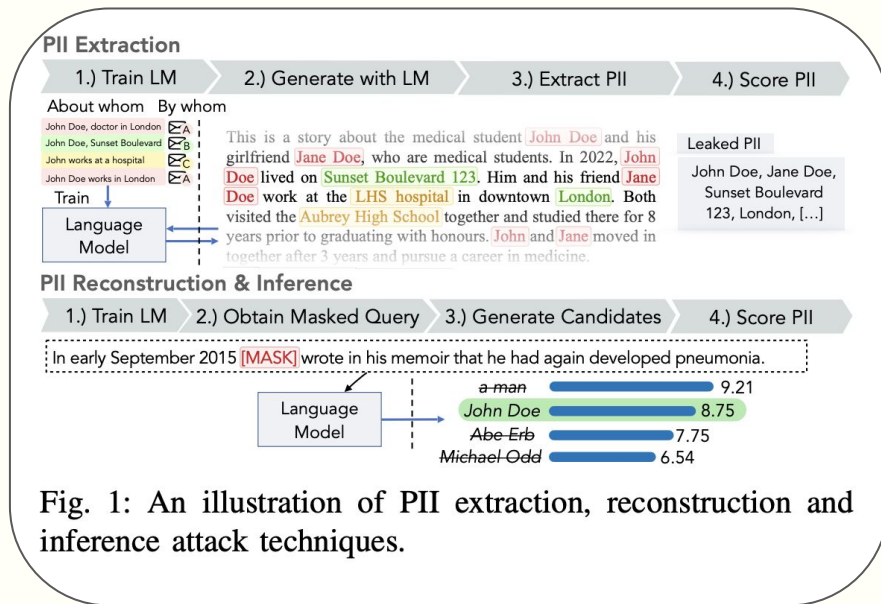
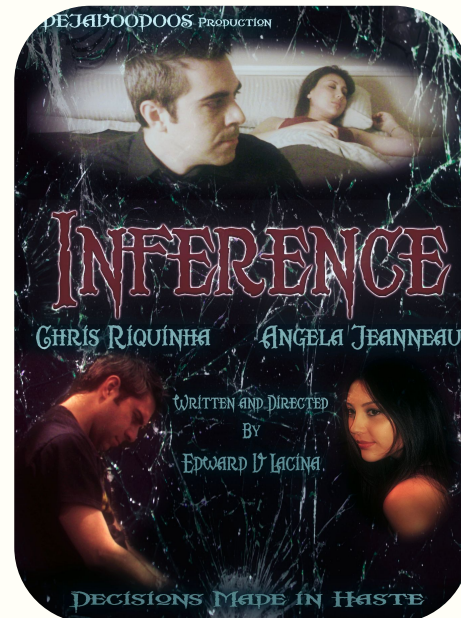


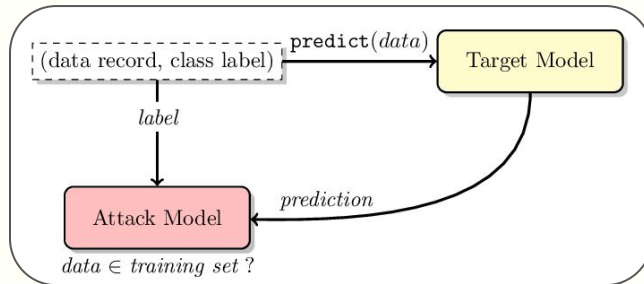
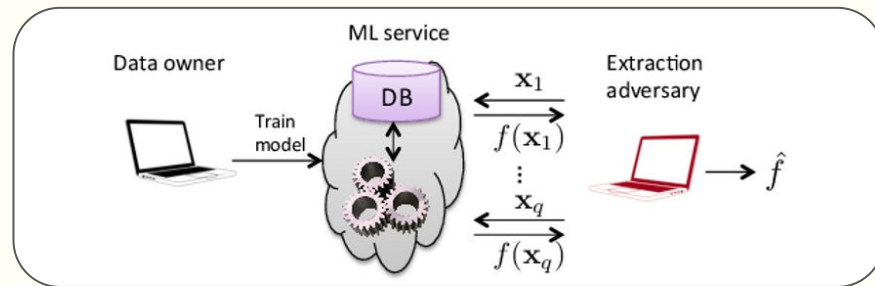
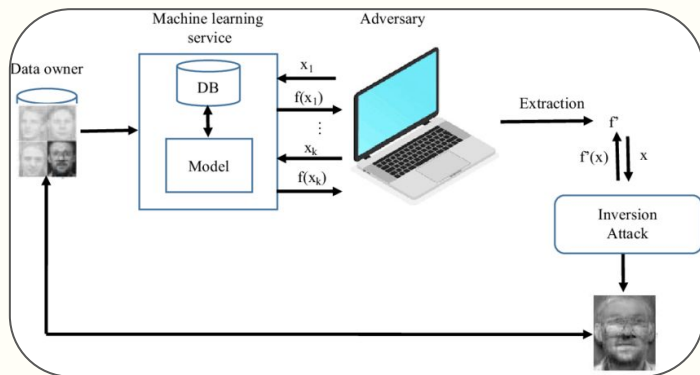
Fig. 1: An illustration of PII extraction, reconstruction and inference attack techniques.

# Extraction, Reconstruction, and Membership Inference



1. [https://m.media-amazon.com/images/M/MV5BNDBkOWNiMTYtMjlkZS00NzJlLTg0ZmUtN2YwYWQ3N2I3N2Y0XkEyXkFqcGdeQXVyMTkxNjUyNQ@@.\\_V1\\_.jpg](https://m.media-amazon.com/images/M/MV5BNDBkOWNiMTYtMjlkZS00NzJlLTg0ZmUtN2YwYWQ3N2I3N2Y0XkEyXkFqcGdeQXVyMTkxNjUyNQ@@._V1_.jpg)
2. [https://m.media-amazon.com/images/M/MV5BYjYyNTQ3NTctOGRkOC00ZTdlWE5NjMtYjdlODdjYmMyODg5XkEyXkFqcGdeQXVyMjgyNjk3MzE@.\\_V1\\_.jpg](https://m.media-amazon.com/images/M/MV5BYjYyNTQ3NTctOGRkOC00ZTdlWE5NjMtYjdlODdjYmMyODg5XkEyXkFqcGdeQXVyMjgyNjk3MzE@._V1_.jpg)
3. [https://m.media-amazon.com/images/M/MV5BMTQ0NDk1Nzk1M15BM15BanBnXkFtZTcwMjU3MTQwNw@@.\\_V1\\_.jpg](https://m.media-amazon.com/images/M/MV5BMTQ0NDk1Nzk1M15BM15BanBnXkFtZTcwMjU3MTQwNw@@._V1_.jpg)

# But Actually...Extraction, Reconstruction, and Membership Inference





# Scrubbing Algorithm

- Iterate over all sentences  $S$
- Extract all candidate PII sequences
- Replace them with **[MASK]**



## Algorithm 1 PII Scrubbing

```
1: procedure SCRUB( $D$ )
2:    $D' \leftarrow \emptyset$ 
3:   for  $S \in D$  do
4:      $\mathcal{C} \leftarrow \text{EXTRACT}(S)$   $\triangleright$  Tag PII with NER
5:     for  $C \in \mathcal{C}$  do
6:        $S_0, S_1 \leftarrow \text{SPLIT}(S, C)$ 
7:        $S \leftarrow S_0$  [MASK]  $S_1$ 
8:        $D' \leftarrow D' \cup \{S\}$ 
9:   return  $D'$ 
```

# Extraction Algorithm

- Sample  $n$  i.i.d. records from  $\mathbf{D}$  to construct a training dataset  $\mathbf{D}$  to train a model  $\theta$
- Adversary is given access to an oracle that returns the probability vector output by  $\theta$  conditioned on arbitrary prefixes of their choosing
- Knowing the number of unique PII sequences  $|\mathbf{C}|$  in  $\mathbf{D}$ , the adversary must produce a set of PII sequences  $\tilde{\mathbf{C}}$  of at most size  $|\mathbf{C}|$

---

**Algorithm 2** PII Extraction

---

```
1: experiment EXTRACTION( $\mathcal{T}, \mathcal{D}, n, \mathcal{A}$ )
2:    $D \sim \mathcal{D}^n$ 
3:    $\theta \leftarrow \mathcal{T}(D)$ 
4:    $\mathcal{C} \leftarrow \bigcup_{S \in \mathcal{D}} \text{EXTRACT}(S)$ 
5:    $\tilde{\mathcal{C}} \leftarrow \mathcal{A}(\mathcal{T}, \mathcal{D}, n, \mathcal{O}_\theta(\cdot), |\mathcal{C}|)$ 

1: procedure  $\mathcal{O}_\theta(S)$ 
2:   return  $\{w \mapsto \Pr(w|S; \theta)\}_{w \in \mathcal{V}}$ 
```

---

# Results

- Scrubbing results in similar perplexities as when training with DP
- Scrubbing and DP degrade the model utility (increase perplexity)

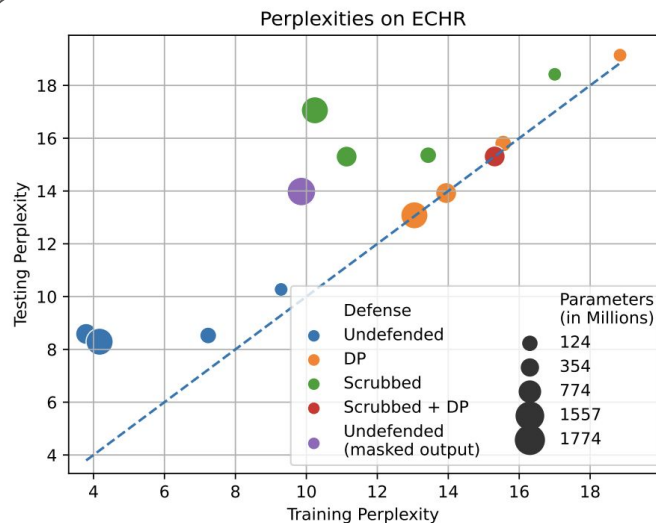


Fig. 2: Utilities of LMs trained (i) undefended, (ii) with scrubbing, (iii) with DP ( $\epsilon = 8$ ), (iv) with scrubbing + DP, and (v) with masked outputs in an ablation study over the LM's size on the ECHR dataset (see Section IV for details).

# Results (continued...)

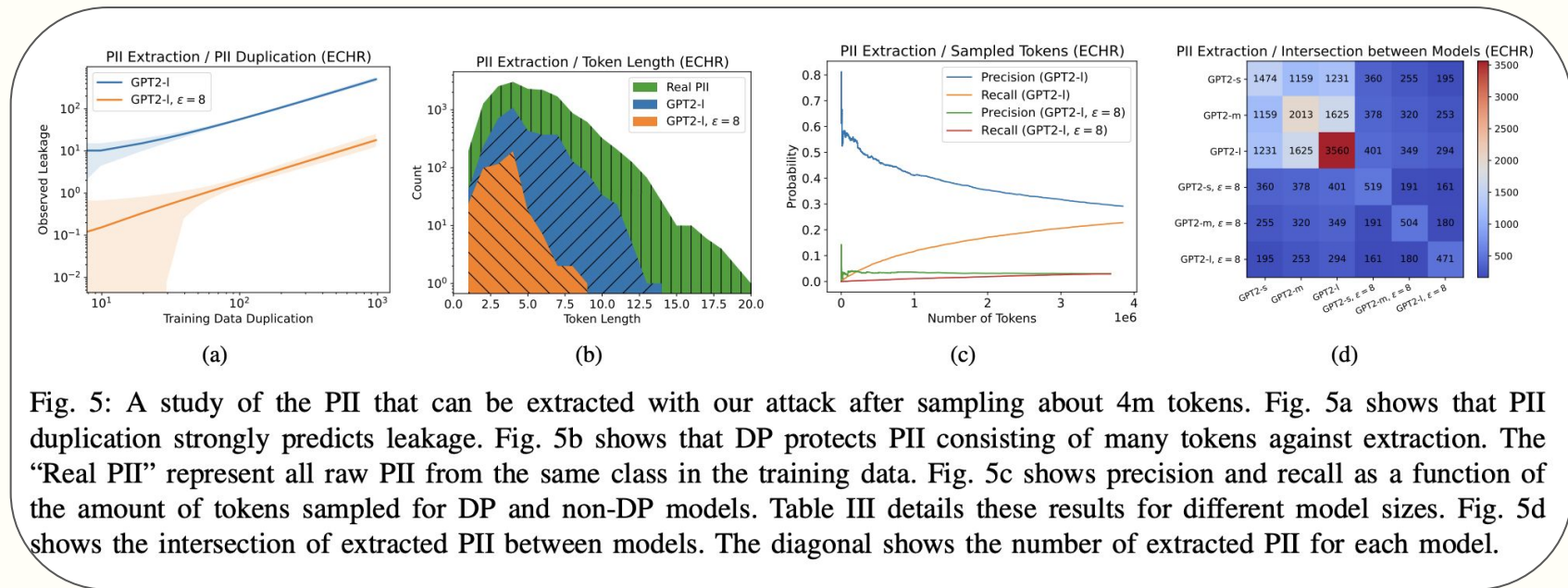




Fig. 5: A study of the PII that can be extracted with our attack after sampling about 4m tokens. Fig. 5a shows that PII duplication strongly predicts leakage. Fig. 5b shows that DP protects PII consisting of many tokens against extraction. The “Real PII” represent all raw PII from the same class in the training data. Fig. 5c shows precision and recall as a function of the amount of tokens sampled for DP and non-DP models. Table III details these results for different model sizes. Fig. 5d shows the intersection of extracted PII between models. The diagonal shows the number of extracted PII for each model.

# Strengths

- First to establish a concrete model on the strengths of defenses against attacks
- Theoretic definition for resistance against data extraction, reconstruction, and membership inference
- Empirically improves upon previous attacks on PII

Default (GPT-3.5)

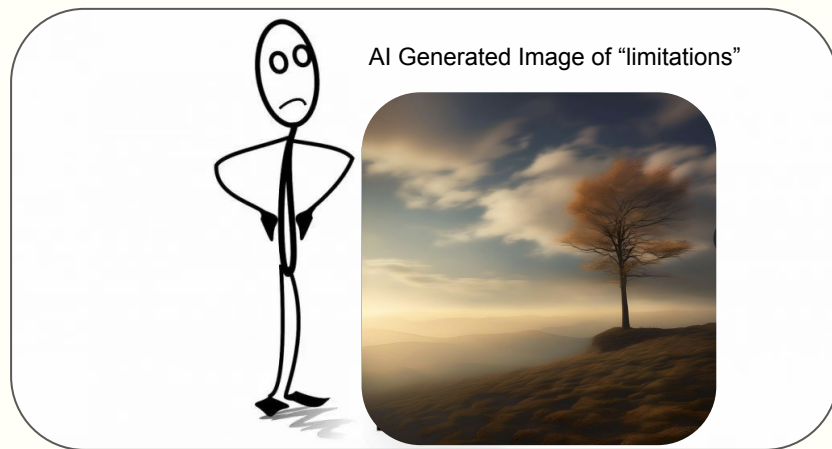
 what are strengths of "Analyzing Leakage of Personally Identifiable Information in Language Models"

 The strengths of analyzing the leakage of Personally Identifiable Information (PII) in language models, as discussed in the research paper or context you are referring to, can vary depending on the specific methodologies and findings of the study. However, in general, such analyses can have several important strengths:

1. Awareness and Accountability: By conducting such analyses, researchers can raise awareness about the potential risks and privacy concerns associated with language models. This can lead to greater accountability among developers, organizations, and policymakers in addressing PII leakage issues.
2. Identifying Vulnerabilities: The research can help identify specific vulnerabilities and mechanisms through which PII can be leaked through language models. This information is crucial for developing countermeasures and mitigation strategies.
3. Ethical Considerations: Analyzing PII leakage provides insights into the ethical implications of using large language models. It can lead to discussions about responsible AI development, data privacy, and informed consent.
4. Public Awareness: Research findings can inform the general public about the risks of sharing sensitive information with language models, which may encourage more responsible data sharing practices.
5. Improved Model Design: The research can guide model developers in improving the design

# Limitations

- Need for better benchmarks (e.g. existing datasets are too small)
- Deployed NER models due to the lack of datasets
- Imagined PII leakage using average-case metrics and did not consider high-precision/low-recall attacks



# Example API Keys

3. **Make API Requests:** You can make API requests to the Alchemy API within your Next.js application. For example:

```
javascript Copy code

import axios from 'axios';

const apiKey = 'YOUR_ALCHEMY_API_KEY';
const apiUrl = 'https://api.alchemyapi.io/v1/some-endpoint';

async function fetchData() {
  try {
    const response = await axios.get(apiUrl, {
      headers: {
        'X-API-Key': apiKey,
      },
      params: {
        // Add any required parameters here
      },
    });
  }
}
```

# Table I

TABLE I: A summary of the difference in threat models between our three PII attacks. (◐ black-box access, ● not available, ○ available)

	Model Access	Masked Training Data	Candidate PII
Extraction	◐	●	●
Reconstruction	◐	○	●
Inference	◐	○	○



# Table II

TABLE II: Summary of Notation

Notation	Description
$\mathcal{T}$	A stochastic training algorithm
$\mathcal{D}$	A distribution over sequences
$\mathcal{E}$	A distribution over PII sequences
$\mathcal{D}^n$	Distribution of $n$ independent sequences from $\mathcal{D}$
$S \sim \mathcal{S}$	Draw a sample $S$ uniformly from a set $\mathcal{S}$
$D \sim \mathcal{D}^n$	Draw $n$ sequences $D$ independently from $\mathcal{D}$
$\mathcal{A}$	A procedure denoting an adversary
$y \leftarrow \mathcal{P}(\vec{x})$	Call $\mathcal{P}$ with arguments $\vec{x}$ and assign result to $y$
$\mathcal{C} \leftarrow \text{EXTRACT}(S)$	Extract the set $\mathcal{C}$ of all PII sequences in $S$
$\mathcal{S} \leftarrow \text{SAMPLE}(S, N, \theta)$	Generate $N$ sequences $\mathcal{S}$ from $\theta$ starting from $S$
$S_0, S_1 \leftarrow \text{SPLIT}(S, \mathcal{C})$	Split $S$ at $\mathcal{C}$ , i.e., $S = S_0 \mathcal{C} S_1$
$S' \leftarrow \text{FILL-MASKS}(S)$	Fill masks in $S$ using a public MLM

# Table II

TABLE II: Summary of Notation

Notation	Description
$\mathcal{T}$	A stochastic training algorithm
$\mathcal{D}$	A distribution over sequences
$\mathcal{E}$	A distribution over PII sequences
$\mathcal{D}^n$	Distribution of $n$ independent sequences from $\mathcal{D}$
$S \sim \mathcal{S}$	Draw a sample $S$ uniformly from a set $\mathcal{S}$
$D \sim \mathcal{D}^n$	Draw $n$ sequences $D$ independently from $\mathcal{D}$
$\mathcal{A}$	A procedure denoting an adversary
$y \leftarrow \mathcal{P}(\vec{x})$	Call $\mathcal{P}$ with arguments $\vec{x}$ and assign result to $y$
$\mathcal{C} \leftarrow \text{EXTRACT}(S)$	Extract the set $\mathcal{C}$ of all PII sequences in $S$
$\mathcal{S} \leftarrow \text{SAMPLE}(S, N, \theta)$	Generate $N$ sequences $\mathcal{S}$ from $\theta$ starting from $S$
$S_0, S_1 \leftarrow \text{SPLIT}(S, \mathcal{C})$	Split $S$ at $\mathcal{C}$ , i.e., $S = S_0 \mathcal{C} S_1$
$S' \leftarrow \text{FILL-MASKS}(S)$	Fill masks in $S$ using a public MLM

Table III

TABLE III: Results for the observed PII extraction on ECHR (top rows), Enron (middle rows), and Yelp-Health (bottom rows) after sampling around 4m tokens across 15k queries.

	GPT2-Small		GPT2-Medium		GPT2-Large	
	No DP	$\epsilon = 8$	No DP	$\epsilon = 8$	No DP	$\epsilon = 8$
<b>ECHR</b>						
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%
<b>Enron</b>						
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%
<b>Yelp-Health</b>						
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%

# Table IV

TABLE II: Summary of Notation

Notation	Description
$\mathcal{T}$	A stochastic training algorithm
$\mathcal{D}$	A distribution over sequences
$\mathcal{E}$	A distribution over PII sequences
$\mathcal{D}^n$	Distribution of $n$ independent sequences from $\mathcal{D}$
$S \sim \mathcal{S}$	Draw a sample $S$ uniformly from a set $\mathcal{S}$
$D \sim \mathcal{D}^n$	Draw $n$ sequences $D$ independently from $\mathcal{D}$
$\mathcal{A}$	A procedure denoting an adversary
$y \leftarrow \mathcal{P}(\vec{x})$	Call $\mathcal{P}$ with arguments $\vec{x}$ and assign result to $y$
$\mathcal{C} \leftarrow \text{EXTRACT}(S)$	Extract the set $\mathcal{C}$ of all PII sequences in $S$
$\mathcal{S} \leftarrow \text{SAMPLE}(S, N, \theta)$	Generate $N$ sequences $\mathcal{S}$ from $\theta$ starting from $S$
$S_0, S_1 \leftarrow \text{SPLIT}(S, \mathcal{C})$	Split $S$ at $\mathcal{C}$ , i.e., $S = S_0 \mathcal{C} S_1$
$S' \leftarrow \text{FILL-MASKS}(S)$	Fill masks in $S$ using a public MLM

# Figure 6

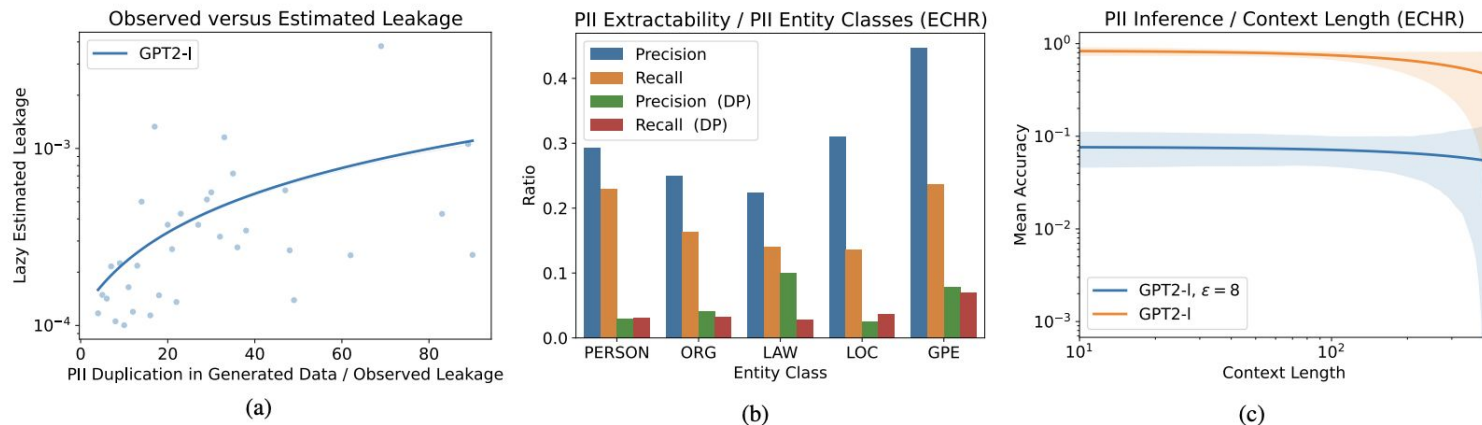
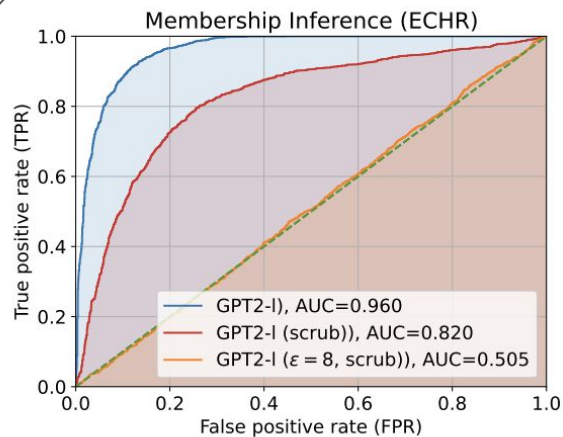
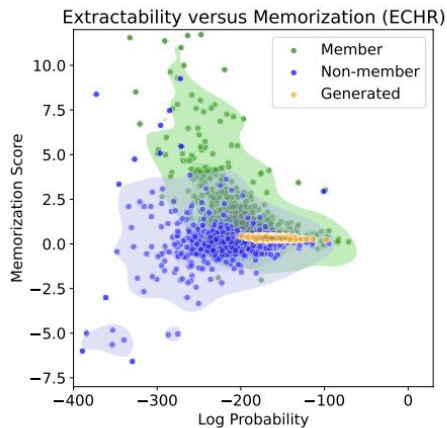


Fig. 6: Fig. 6a shows the correlation between the observed and estimated leakage. Fig. 6b shows the precision and recall for other entity classes on the ECHR dataset. Fig. 6c shows the mean inference accuracy relative to the context length, which is the length of the combined prefix and suffix for a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

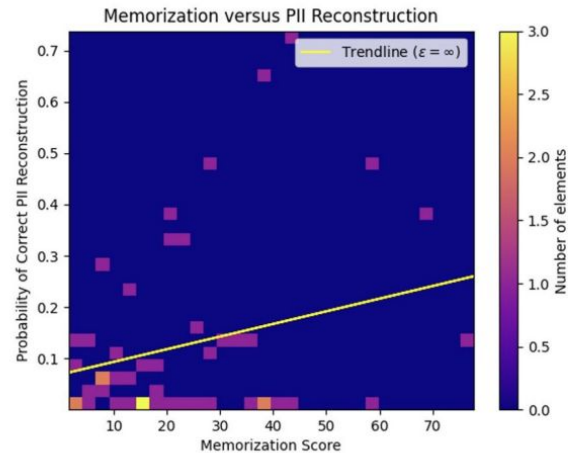
# Figure 7



(a)



(b)



(c)

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.

# Perplexity

$$\text{PPL}(w_1, \dots, w_n; \theta) = \exp \left( -\frac{1}{n} \sum_{i=1}^n \log \Pr(w_i | w_1, \dots, w_{i-1}; \theta) \right)$$