Alpaca against Vicuna:
Using LLMs to Uncover Memorization of LLM

Aly M. Kassem, Omar Mahmoud, Niloofar Mireshghallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, Santu Rana
Summer 2024

@niloofar_mire
ACT I: What is memorization and regurgitation?

“Don’t repeat this...”
Memorization and Regurgitation

Researchers recovered over **10,000 examples**, including a dozen PII, from ChatGPT’s training data at a query cost of **$200 USD**
Memorization and Regurgitation

Not just LLMs!

Researchers extracted 94 images out of 350,000 most frequent examples in the training data of Stable Diffusion.
Memorization and Regurgitation

Not a recent problem!

This xkcd cartoon is from June 2019!
DIY Extraction

- Github Co-pilot:

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hi everyone, my name is Anish Athalye and I'm a PhD student at Stanford University.</td>
</tr>
</tbody>
</table>
DIY Extraction

• Github Co-pilot:

Title:
Hi everyone, my name is Anish Athalye and I'm a PhD student at Stanford University.

https://www.anish.io

Anish Athalye
I am a PhD student at MIT in the PDOS group. I'm interested in formal verification, systems, security, and machine learning.

GitHub: @anishathalye   Blog: anishathalye.com
ACT II: Why should we care?

“Honey, why does the toaster know it’s my birthday tomorrow?”
What data are models trained on?

We are running out of open data!

We aren’t running out of training data, we are running out of open training data

Data licensing deals, scaling, human inputs, and repeating trends in open vs. closed LLMs.

For months we’ve been getting stories about how the leading teams training language models (LMs) are running out of data for their next generation of models – vaguely insinuating a struggle for big tech’s darling industry with no strategic claims beyond the fact that the second derivative on training dataset size is negative.
What data are models trained on?

We are running out of open data!

ChatGPT has approximately 100 million monthly active users, let’s call it 10 million daily queries into ChatGPT, of which the average answer is 1000 tokens. This puts them at 10 billion candidate tokens to retrain their models every single day. Not all of this is valuable, and as little as possible will be released, but if they really need more places to look for text data, they have it.
What does user data look like?

- WildChat is a dataset of human-LLM conversations in the ‘wild’.
- Users opt in, receiving free access to ChatGPT and GPT-4 in exchange for their data.
Trust No Bot? Personal Disclosures in Human-LLM Conversations
Niloofar Mireshghallah,* Maria Antoniak,* Yash More,* Yejin Choi, Golnoosh Farnadi

*On Arxiv soon!

Breaking News: Case Studies of Generative AI's Use in Journalism
Natalie Grace Brigham, Chongjiu Gao, Tadayoshi Kohno, Franziska Roesner, Niloofar Mireshghallah

https://arxiv.org/abs/2406.13706
What types of sensitive data is in there?

<table>
<thead>
<tr>
<th>sensitive topic</th>
<th>academic and education info</th>
<th>fandom</th>
<th>financial and corporate info</th>
<th>healthcare information</th>
<th>job, visa, and other applications</th>
<th>quoted code</th>
<th>sexual and erotic content</th>
<th>user's emotions and mental health</th>
<th>user's hobbies and habits</th>
<th>user's personal relationships</th>
<th>user's politics and religion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.74 0.18 0.03 0.1 0.19 0.42 0.26 0.12 0.53 0.24 0.067 0.13 0.023 0.75 0.077 0.041 0.47 0.29</td>
<td>0.12 0.003 0.012 0.13 0.062 0.02 0.013 0.045 0.051 0.19 0.026 0.49 0.022 0.28 0.53 0.12 0.058</td>
<td>0.085 0.96 0.0059 0.012 0.082 0.11 0.054 0.15 0.593 0.076 0.037</td>
<td>0.0076 0.12 0.026 0.0092 0.077 0.074</td>
<td>0.012 0.013 0.006 0.035 0.082 0.021 0.17 0.12 0.233 0.22 0.015</td>
<td>0.073 0.013 0.96 0.48 0.024 0.011 0.23 0.0044 0.011 0.047 0.015</td>
<td>0.029</td>
<td>0.027 0.016 0.022 0.008 0.021 0.43 0.16 0.38</td>
<td>0.027 0.0086 0.061 0.0053 0.0069 0.052 0.45 0.03</td>
<td>0.051 0.014 0.0096 0.016</td>
<td>0.012 0.17 0.0089 0.022 0.23</td>
</tr>
</tbody>
</table>
What types of sensitive data is in there?
What types of sensitive data is in there?
<table>
<thead>
<tr>
<th>Task</th>
<th>NLP Score</th>
<th>Raw Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>answering multiple choice questions</td>
<td>0.049</td>
<td>0.24</td>
</tr>
<tr>
<td>brainstorming and generating ideas</td>
<td>0.021</td>
<td>0.27</td>
</tr>
<tr>
<td>code editing and debugging</td>
<td>0.003</td>
<td>0.22</td>
</tr>
<tr>
<td>code generation</td>
<td>0.002</td>
<td>0.21</td>
</tr>
<tr>
<td>comparison, ranking, and recommendation</td>
<td>0.024</td>
<td>0.26</td>
</tr>
<tr>
<td>editing existing text</td>
<td>0.003</td>
<td>0.24</td>
</tr>
<tr>
<td>explanation, how-to, practical advice</td>
<td>0.002</td>
<td>0.21</td>
</tr>
<tr>
<td>generating communications</td>
<td>0.036</td>
<td>0.47</td>
</tr>
<tr>
<td>generating non-fictional documents</td>
<td>0.016</td>
<td>0.32</td>
</tr>
<tr>
<td>information retrieval</td>
<td>0.017</td>
<td>0.25</td>
</tr>
<tr>
<td>model jailbreaking</td>
<td>0.0075</td>
<td>0.56</td>
</tr>
<tr>
<td>personal advice</td>
<td>0.0075</td>
<td>0.55</td>
</tr>
<tr>
<td>role-playing</td>
<td>0.0075</td>
<td>0.55</td>
</tr>
<tr>
<td>solving logic, math, and word problems</td>
<td>0.047</td>
<td>0.47</td>
</tr>
<tr>
<td>song and poem generation</td>
<td>0.033</td>
<td>0.33</td>
</tr>
<tr>
<td>story and script generation</td>
<td>0.0002</td>
<td>0.04</td>
</tr>
<tr>
<td>summarization</td>
<td>0.029</td>
<td>0.34</td>
</tr>
<tr>
<td>translation</td>
<td>0.015</td>
<td>0.3</td>
</tr>
</tbody>
</table>

What types of PII do we see?
What types of PII do we see?

Example: This letter is to confirm that I, Zxxx Qxxx, am the daughter of Qxxxxx Qxxx .... I will begin my course in Engineering Science as a first-year student at Oxford University in October. My passport number is EJxxxxxxx0, and my student visa number is xxxxxxxx00...
What types of PII do we see?

Example: This letter is to confirm that I, Zxxx Qxxx, am the daughter of Qxxxxx Qxxx. I will begin my course in Engineering Science as a first-year student at Oxford University in October. My passport number is EJxxxxxx0, and my student visa number is xxxxxx00...
Example Query to ChatGPT— WhatsApp conversation

“Hello I am a Lovin Malta journalist and one woman contacted me regarding an issue she has with the government and other stuff that the government does not provide for her child who is disabled. anaylise the whatsapp convo and write an article out of it. tell me if you need more information that would help give the article the human element:
"Hello, I am a Lovin Malta journalist and one woman contacted me regarding an issue she has with the government and other stuff that the government does not provide for her child who is disabled. Analyse the WhatsApp convo and write an article out of it. Tell me if you need more information that would help give the article the human element:"
Example Query to ChatGPT—WhatsApp conversation

[10:48, 06/04/2023] <PHONE_NUMBER>: no I would not like my children’s photos on the article

[10:49, 06/04/2023] <PHONE_NUMBER>: And re conditions I will only mention the one who needs to travel overseas as it’s the only one that is a visible disability cos he cannot walk

[11:23, 06/04/2023] <PHONE_NUMBER>: I have 3 children, one is 8 and the other 2 are 4 years old, once one of our 4 year old was diagnosed with PVL a brain condition resulting in Cerebral palsy I found myself in a new community in Malta that is of parents with children with disabilities who in my opinion is not supported enough in Malta.

[12:38, 06/04/2023] <PRESIDIO_ANONYMIZED_PHONE_NUMBER>: If u feel my voice is enough and no need for others at this point leave it as me only

[14:40, 06/04/2023] <PRESIDIO_ANONYMIZED_PHONE_NUMBER>: Audrey Jones

[14:40, 06/04/2023] <PRESIDIO_ANONYMIZED_PHONE_NUMBER>: This mother is also interested to share info
Example Query to ChatGPT— WhatsApp conversation

"Hello I am a Lovin Malta journalist and one woman contacted me regarding an issue she has with the government and other stuff that the government does not provide for her child who is disabled. Analyse this conversation and write an article out of it. Tell me if you need more information that would help give the article the human element."
"Hello I am a Lovin Malta journalist and one woman contacted me regarding an issue she has with the government and other stuff that the government does not provide for her child who is disabled. Can you analyse the WhatsApp conversation and write an article out of it? Tell me if you need more information to give the article the human element."

Average ROUGE-L of 0.62 for published articles
Leakage of this data, either through memorization or data breaches, can have huge ramifications!
ACT III: How do we formalize memorization in LLMs?
Membership Inference Attacks

- An **upper bound on leakage** is measured by mounting a **membership inference attack (MIA)**.
- Can an adversary infer whether a **particular data point “x”** is part of the **training set**?

Target sample (x)
Membership Inference Attacks

- An upper bound on leakage is measured by mounting a **membership inference attack (MIA)**.
- Can an adversary infer whether a **particular data point** “x” is part of the **training set**?

![Diagram](image)

Mr. Smith has lung Cancer.

Target sample (x)

Training Data

✔ Member

✗ Non-member

Mireshghallah et al. “Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks”, EMNLP 2022
Membership Inference Attacks

- An upper bound on leakage is measured by mounting a membership inference attack (MIA).
- Can an adversary infer whether a particular data point \( x \) is part of the training set?

The success rate of the attack is a measure of leakage.

Miresghallah et al. “Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks”, EMNLP 2022
Membership Inference or …?

Do Membership Inference Attacks Work on Large Language Models?

Michael Duan 1 2 Anshuman Suri 1 2 Niloofar Miresghallah 1 Sewon Min 1 Weijia Shi 1 Luke Zettlemoyer 1 Yulia Tsvetkov 1 Yejin Choi 1 3 David Evans 2 Hannaneh Hajishirzi 1 3

Abstract

Membership inference attacks (MIAs) attempt to predict whether a particular datapoint is a member of a target model’s training data. Despite extensive research on traditional machine learning models, there has been limited work studying MIA on the pre-training data of large language models (LLMs). We perform a large-scale evaluation of MIAs over a suite of language models (LMs) trained on the Pile, ranging from 160M to 12B

blind baselines beat membership inference attacks for foundation models

Debeshee Das Jie Zhang Florian Tramèr

ETH Zurich

Abstract

Membership inference (MI) attacks try to determine if a data sample was used to train a machine learning model. For foundation models trained on unknown Web data, MI attacks can be used to detect copyrighted training materials, measure test set contamination, or audit machine unlearning. Unfortunately, we find that evaluations of MI attacks for foundation models are flawed, because they sample members and non-members from different distributions. For 8 published MI evaluation datasets, we show that blind attacks—that distinguish the member and non-member distributions without looking at any trained model—outperform state-of-the-art MI attacks. Existing evaluations thus tell us nothing about membership leakage of a foundation model’s training data.
Membership Inference or …?

Do Membership Inference Attacks Work?

Michael Duan 1, Anshuman Suri 1,2, Niloofar Mobasseri 1, Luke Zettlemoyer 1, Yulia Tsvetkova 1, Yejin Choi 1

Abstract

Membership inference attacks (MIAs) attempt to predict whether a particular datapoint is a member of a target model’s training data. Despite extensive research on traditional machine learning models, there has been limited work studying MIA on the pre-training data of large language models (LLMs). We perform a large-scale evaluation of MIAs over a suite of language models (LMs) trained on the Pile, ranging from 160M to 12B parameters. We find a significant trend of increasing MIA success on more massive models. We also show that pre-trained LLMs exhibit evidence of memorization on the pre-training data, via text that matches the training set. These findings challenge the assumptions of existing studies that MIA is not a threat to modern language models. Finally, we release the most effective MIA tool, MIA-X, which not only achieves state-of-the-art results but also can be easily reproduced and adapted to future LLMs.

Controversial take: This is exactly why we should retire membership inference for the really large models, and look at more direct and concrete evidence of memorization. Such as training data extraction and deja vu (arxiv.org/abs/2304.13850).

Florian Tramèr @florian_tramer · Jun 25

We’re releasing the strongest membership inference attack for foundation models.

Our attack applies to LLMs, vLMs, CLIP, Diffusion models and is SOTA on all.

Show more

Show this thread

<table>
<thead>
<tr>
<th>MI dataset</th>
<th>Metric</th>
<th>Best Reported</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>WikiMIA</td>
<td>TPR@5%FPR</td>
<td>43.2%</td>
<td>94.4%</td>
</tr>
<tr>
<td>BookMIA</td>
<td>AUC ROC</td>
<td>88.0%</td>
<td>90.5%</td>
</tr>
<tr>
<td>Temporal Wiki</td>
<td>AUC ROC</td>
<td>79.6%</td>
<td>79.9%</td>
</tr>
<tr>
<td>Temporal arXiv</td>
<td>AUC ROC</td>
<td>72.3%</td>
<td>73.1%</td>
</tr>
<tr>
<td>ArXiv-1 month</td>
<td>TPR@1%FPR</td>
<td>5.9%</td>
<td>13.4%</td>
</tr>
<tr>
<td>Multi-Webdata</td>
<td>TPR@1%FPR</td>
<td>40.3%</td>
<td>83.5%</td>
</tr>
<tr>
<td>LAION-MI</td>
<td>TPR@1%FPR</td>
<td>2.5%</td>
<td>9.9%</td>
</tr>
<tr>
<td>Gutenberg</td>
<td>TPR@1%FPR</td>
<td>18.8%</td>
<td>59.6%</td>
</tr>
</tbody>
</table>

Current Membership Inference Attacks for Foundation Models

Jie Zhang Florian Tramèr

ETH Zurich

Abstract

Membership inference attacks try to determine if a data sample was used to train a machine learning model. Existing studies, however, are not adequate to judge whether new models trained on unknown Web data, MI attacks can be used to infer which samples are members from different distributions. For 8 published MI evaluation setups that distinguish the member and non-member distributions, we show that variants of our attack’s model—outperform state-of-the-art MI attacks. Existing evaluations thus tell us nothing about membership leakage of a foundation model’s training data.
Extractability!

Extractability: A sequence $s$ of length $N$ is extractable from a model $h$ if there exists a prefix $c$ such that:

$$s \leftarrow \arg \max_{s'} h(s' \mid c), \quad \text{such that} \quad |s'| = N$$

Example: the email address “alice@wonderland.com” is extractable if prompting the model with “Their email address is...” and decoding from it yields “alice@wonderland.com” as the most probable output.

Rethinking LLM Memorization through the Lens of Adversarial Compression

Avi Schwarzschild
avr@andrew.cmu.edu
Carnegie Mellon University

Zhilif Feng
zhilif@andrew.cmu.edu
Carnegie Mellon University

Pratyush Maini
pratyushmaini@cmu.edu
Carnegie Mellon University

Zachary C. Lipton
zachary@andrew.cmu.edu
Carnegie Mellon University

J. Zico Kolter
jzk@cs.cmu.edu
Carnegie Mellon University

Abstract

Large language models (LLMs) trained on web-scale datasets raise substantial concerns regarding permissible data usage. One major question is whether these models “memorize” all their training data or they integrate many data sources in some way more akin to how a human would learn and synthesize information? The answer hinges, to a large degree, on how we define memorization. In this work, we propose the Adversarial Compression Ratio (ACR) as a metric for assessing memorization in LLMs—a given string from the training data is considered memorized if it can be elicited by a prompt shorter than the string itself. In other words, these strings can be “compressed” with the model by computing adversarial prompts of fewer tokens. We outline the limitations of existing notions of memorization and show how the ACR overcomes these challenges.

Recite, Reconstruct, Recollect:
Memorization in LMs as a Multifaceted Phenomenon

USVSN Sai Prashanth$^{* , 1}$ Alvin Deng$^{* , 1 , 4}$ Kyle O’Brien$^{* , 1 , 2}$ Jyothir S V$^{* , 1 , 3}$
Mohammad Aflah Khan$^{1 , 6}$ Jaydeep Borkar$^{5}$
Christopher A. Choquette-Choo$^{6}$ Jacob Ray Fuehne$^{8}$ Stella Biderman$^{1}$
Tracy Ke$^{7}$ Katherine Lee$^{7}$, 9 Naomi Saphra$^{9 , 10}$

$^{1}$EleutherAI $^{2}$Microsoft $^{3}$New York University $^{4}$DatologyAI $^{5}$Northeastern University
$^{6}$Indraprastha Institute of Information Technology Delhi $^{7}$Google DeepMind
$^{8}$University of Illinois at Urbana-Champaign $^{9}$Harvard University $^{10}$Kemper Institute

Correspondence: katherinlee@gmail.com and nsaphra@fas.harvard.edu

Abstract

Memorization in language models is typically treated as a homogenous phenomenon, neglecting the specifics of the memorized data. We instead model memorization as the effect of a set of complex factors that describe each sample and relate it to the model and corpus. To build intuition around these factors, we break memorization down into a taxonomy: recitation of highly duplicated sequences, reconstruction of inherently predictable sequences, and recollection of sequences that are neither. We

Our taxonomy, illustrated in Fig. 1, defines three types of LM memorization based on colloquial descriptions of human memorization. Humans recite direct quotes that they commit to memory through repeated exposure, so LMs recite highly duplicated sequences. Humans reconstruct a passage by remembering a general pattern and filling in the gaps, so LMs reconstruct inherently predictable boiler-plate templates. Humans sporadically recollect an episodic memory or fragment after a single exposure, so LMs recollect other sequences seen rarely during training.
In this talk, we focus on extractability!
Extractability

Extractability: A sequence $s$ of length $N$ is extractable from a model $h$ if there exists a prefix $c$ such that:

$$s \leftarrow \arg \max_{s'} h(s' | c), \quad \text{such that} \quad |s'| = N$$

If the prefix $c$ is part of the original prefix of $s$ in the training data, then sequence $s$ is called discoverable.

We will call this the prefix-suffix (P-S) from this point on.
Relaxations to Exact String Matching

- Huang et al. (2023) consider **ROUGE-L > 0.5** as successful extraction
- Ippolito et al. (2022) consider **BLEU > 0.75** as a successful extraction
- Biderman et al. (2023) report a memorization score based on the **longest common subsequence match** with the ground truth (equivalent to the ROUGE-L score):

<table>
<thead>
<tr>
<th>Prompt</th>
<th>True Continuation</th>
<th>Greedily Generated Sequence</th>
<th>Memorization Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>The patient name is Jane Doe and she lives in the United States.</td>
<td>John Doe and he lives in the United Kingdom</td>
<td>0+1+1+0+1+1+1+0+1+1 = 0.7</td>
<td></td>
</tr>
<tr>
<td>Pi is defined as the ratio of the radius of a circle to its</td>
<td>a famous decimal that never enters a repeating pattern.</td>
<td>0+0+0+0+0+0+0+0+0+0+0 = 0</td>
<td></td>
</tr>
<tr>
<td>The case defendant is Billy Bob. They are on trial for tax fraud</td>
<td>Billy Bob. Are they really on trial for tax fraud</td>
<td>1+1+1+0+1+0+0+0+1+0+1 = 0.3</td>
<td></td>
</tr>
<tr>
<td>The case defendant is Billy Bob. They are on trial for tax fraud</td>
<td>Billy Bob. They are on trial for tax fraud</td>
<td>1+1+1+1+1+1+1+1+1+1+1 = 1</td>
<td></td>
</tr>
</tbody>
</table>

The memorization score is calculated as:

$$\text{score}(M, N) = \frac{1}{N} \sum_{i}^{N} 1(S_{M+i} = G_{M+i})$$

Where $G$ is the model’s **greedily generated** sequence and $S$ is the dataset’s **true continuation** on a given prompt, and $N$ is the **length** of the **true continuation** and greedily generated sequence, and $M$ is the **length** of the **prompt**.
What is missing?
Memorization in instruction-tuned models

• There is no study of memorization specific to instruction tuned models, comparing against their base models, even using prefix-suffix!
Memorization in instruction-tuned models

• There is no study of memorization specific to instruction tuned models, comparing against their base models, even using prefix-suffix!

• Current prefix-suffix baseline is not adversarial: Maybe we can do better? Maybe the training data is not the upper-bound context to elicit memorized pre-training data
Memorization in instruction-tuned models

• There is no study of memorization specific to instruction tuned models, comparing against their base models, even using prefix-suffix!

• Current prefix-suffix baseline is not adversarial: Maybe we can do better? Maybe the training data is not the upper-bound context to elicit memorized pre-training data

• Current prefix-suffix baseline is not tailored for instruction tuned models: Maybe there is a distribution shift, it may not be uncovering memorization as well as it does in the base models

We set out to answer these questions, by proposing a prompt optimization method targeting extraction!
ACT IV: Let’s do prompt optimization!
Optimization Problem

Consider a sequence $d \in D$, where $D$ is the pre-training dataset of a model $M$. 
Optimization Problem

• Consider a sequence $d \in D$, where $D$ is the pre-training dataset of a model $M$.

• The objective is to find an input prompt $p^*$ that maximizes the overlap between the output sequence of the model $M(p^*)$ and $d$:

$$p^* = \arg \max_p \mathcal{O}_{d,M}(p)$$
Optimization Problem

• Consider a sequence $d \in D$, where $D$ is the pre-training dataset of a model $M$.

• The objective is to find an input prompt $p^*$ that maximizes the overlap between the output sequence of the model $M(p^*)$ and $d$:

$$p^* = \arg\max_p \mathcal{O}_{d,M}(p)$$

Where $\mathcal{O}_{d,M}(p)$ can be:

1. $\mathcal{O}_{d,M}(p) = LCS(M(p), d_{\text{suffix}})$: Maximize the overlap between generation from model $M$ given prompt $p$ and the suffix.
Optimization Problem

• Consider a sequence \( d \in D \), where \( D \) is the pre-training dataset of a model \( M \).

• The objective is to find an input prompt \( p^* \) that maximizes the overlap between the output sequence of the model \( M(p^*) \) and \( d \):

\[
p^* = \arg\max_p \mathcal{O}_{d,M}(p)
\]

Where \( \mathcal{O}_{d,M}(p) \) can be:

1. \( \mathcal{O}_{d,M}(p) = LCS(M(p), d_{\text{suffix}}) \): Maximize the overlap between generation from model \( M \) given prompt \( p \) and the suffix.

2. \( \mathcal{O}_{d,M}(p) = \alpha \cdot LCS(M(p), d_{\text{suffix}}) + (1 - \alpha) \cdot -LCS(p, d_{\text{suffix}}) \): Maximize the overlap mentioned above, while minimizing the overlap between the prompt and the suffix.
Proposed Optimization Algorithm

Algorithm 1 Interactive Sampling Algorithm

1: **Input:** pre-training sample $d$, $M$, $M'$, $M_{init}$
2: $p_{init} \leftarrow M_{init}(d)$  
   
   //Construct initial prompt
3: $p_{t-1} \leftarrow p_{init}$

Given a paragraph snippet, please generate a question that asks for the generation of the paragraph.
Proposed Optimization Algorithm

Algorithm 1 Interactive Sampling Algorithm

\begin{align*}
\text{1: Input:} & \quad \text{pre-training sample } d, M, M', M_{\text{init}} \\
\text{2: } & \quad p_{\text{init}} \leftarrow M_{\text{init}}(d) \\
\text{3: } & \quad p_{t-1} \leftarrow p_{\text{init}} \\
\end{align*}

//Construct initial prompt

Given a paragraph snippet, please generate a question that asks for the generation of the paragraph.

Goal is to turn the statement into an instruction!
Proposed Optimization Algorithm

Algorithm 1 Interactive Sampling Algorithm

1: **Input:** pre-training sample $d$, $M$, $M'$, $M_{\text{init}}$
2: $p_{\text{init}} \leftarrow M_{\text{init}}(d)$  //Construct initial prompt
3: $p_{t-1} \leftarrow p_{\text{init}}$
4: **for** $t = 3$ **do**
5: \[ p_t \sim M'(Instr|p_{t-1}, n = 24) \]  //Sample 24
6: \[
O = \alpha \cdot \text{LCS}(M(p_t), d_{\text{suffix}}) + (1 - \alpha) \cdot \\
-\text{LCS}(p_t, d_{\text{suffix}})
\]
7: $p_t = \text{arg max}(O)$  //Obtain the highest scoring prompt

Rejection sampling:
Proposed Optimization Algorithm

Algorithm 1 Interactive Sampling Algorithm

1: **Input:** pre-training sample $d$, $M$, $M'$, $M_{\text{init}}$
2: $p_{\text{init}} \leftarrow M_{\text{init}}(d)$  //Construct initial prompt
3: $p_{t-1} \leftarrow p_{\text{init}}$
4: for $t = 3$ do
5: $p_t \sim M'(\text{Instr}|p_{t-1}, n = 24)$  //Sample 24
6: $O = \alpha \cdot \text{LCS}(M(p_t), d_{\text{suffix}}) + (1 - \alpha) \cdot -\text{LCS}(p_t, d_{\text{suffix}})$
7: $p_t = \text{arg max}(O)$  //Obtain the highest scoring prompt

Proposals generated by 'attacker model'
Proposed Optimization Algorithm

**Algorithm 1** Interactive Sampling Algorithm

1: **Input:** pre-training sample $d$, $M$, $M'$, $M_{\text{init}}$
2: $p_{\text{init}} \leftarrow M_{\text{init}}(d)$ //Construct initial prompt
3: $p_{t-1} \leftarrow p_{\text{init}}$
4: for $t = 3$ do
5: \[ p_t \sim M'(\text{Instr}(p_{t-1}, n = 24)) \] //Sample 24
6: \[ O = \alpha \cdot \text{LCS}(M(p_t), d_{\text{sufffix}}) + (1 - \alpha) \cdot -\text{LCS}(p_t, d_{\text{sufffix}}) \]
7: $p_t = \text{arg max}(O)$ //Obtain the highest scoring prompt

I will provide you with previous questions. Write your new question by paraphrasing the old ones.
Proposed Optimization Algorithm

Algorithm 1 Interactive Sampling Algorithm

1: **Input**: pre-training sample $d$, $M$, $M'$, $M_{\text{init}}$
2: $p_{\text{init}} \leftarrow M_{\text{init}}(d)$  //Construct initial prompt
3: $p_{t-1} \leftarrow p_{\text{init}}$
4: **for** $t = 3$ **do**
5: $p_t \sim M'(\text{Instr}|p_{t-1}, n = 24)$  //Sample 24
6: $\mathcal{O} = \alpha \cdot \text{LCS}(M(p_t), d_{\text{suff}}) + (1 - \alpha) \cdot$\text{LCS}(p_t, d_{\text{suff}})$  \text{//Obtain the highest scoring prompt}$
7: p_t = \text{arg max}(\mathcal{O})$
Proposed Optimization Algorithm

**Algorithm 1 Interactive Sampling Algorithm**

1: **Input:** pre-training sample $d$, $M$, $M'$, $M_{\text{init}}$
2: $p_{\text{init}} \leftarrow M_{\text{init}}(d)$ \hspace{1cm} //Construct initial prompt
3: $p_{t-1} \leftarrow p_{\text{init}}$
4: **for** $t = 3$ **do**
5: \begin{align*}
& p_t \sim M'(\text{Instr}|p_{t-1}, n = 24) \hspace{1cm} //\text{Sample 24} \\
& \mathcal{O} = \alpha \cdot \text{LCS}(M(p_t), d_{\text{suff}}) + (1 - \alpha) \cdot \\
& \quad \quad -\text{LCS}(p_t, d_{\text{suff}})
\end{align*}
6: \begin{align*}
& p_t = \text{arg max}(\mathcal{O}) \hspace{1cm} //\text{Obtain the highest scoring prompt}
\end{align*}

Best of n
Proposed Optimization Algorithm

Algorithm 1 Interactive Sampling Algorithm

1. **Input:** pre-training sample \( d, M, M', M_{\text{init}} \)
2. \( p_{\text{init}} \leftarrow M_{\text{init}}(d) \)  
   //Construct initial prompt
3. \( p_{t-1} \leftarrow p_{\text{init}} \)
4. **for** \( t = 3 \) **do**
   5. \( p_t \sim M'(Instr|p_{t-1}, n = 24) \)  
      //Sample 24
   6. \( \mathcal{O} = \alpha \cdot \text{LCS}(M(p_t), d_{\text{suffix}}) + (1 - \alpha) \cdot \text{LCS}(p_t, d_{\text{suffix}}) \)
   7. \( p_t = \arg \max(\mathcal{O}) \)  
      //Obtain the highest scoring prompt
5. **end for**
6. \( p^* = \arg \max(p_0, \ldots, p_t) \)  
   //get the highest over iters
7. **return** \( p^* \)  
   //Return optimal prompt

Best of all
Optimization Procedure

Initial Prompt $p_{init}$

Generate a code snippet that serves as a header comment for...
Optimization Procedure

Initial Prompt $p_{init}$

Generate a code snippet that serves as a header comment for...

Generated Completion

that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.
2. Redistributions in binary...
Optimization Procedure

**Initial Prompt** $p_{init}$

Generate a code snippet that serves as a header comment for...

$LCS(p_{init}, d_{suffix}): 0.08 (\downarrow)$

**Generated Completion**

that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary

$LCS(M(p_{init}), d_{suffix}): 0.21 (\uparrow)$

Victim LLM $M$
I have a question in file, How should I properly structure the header

Optimization Procedure

Initial Prompt $p_{init}$

Generate a code snippet that serves as a header comment for...

$\text{LCS}(p_{init}, d_{suffix}): 0.08 (\downarrow)$

Generated Completion

that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
2. Redistributions in binary format:

$L\text{CS}(M(p_{init}), d_{suffix}): 0.21 (\uparrow)$

Attacker LLM $M'$

Victim LLM $M$

Optimized Prompt $p^*$
Optimization Procedure

Initial Prompt $p_{init}$

Generate a code snippet that serves as a header comment for...

$LCS(p_{init}, d_{suffix})$: 0.08 (↓)

Generated Completion

that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer.

$LCS(M(p_{init}), d_{suffix})$: 0.21 (↑)

Attacker LLM $M'$

I have a question in file, How should I properly structure the header

Optimized Prompt $p^*$

Generated Completion

# MyRobotLab - Copyright (C) 2023 Author Name
<author@example.com>

This file is part of MyRobotLab.

version 3 of the License, or (at your option) any later version. This program is...

Victim LLM $M$
Optimization Procedure

**Initial Prompt** $p_{\text{init}}$

Generate a code snippet that serves as a header comment for...

$LCS(p_{\text{init}},d_{\text{suffix}}): 0.08 (\downarrow)$

**Generated Completion**

that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

$LCS(M_{\text{init}},d_{\text{suffix}}): 0.21 (\uparrow)$

**Attacker LLM** $M'$

I have a question in file, How should I properly structure the header

Generated Completion

# MyRobotLab - Copyright (C) 2023 Author Name
<author@example.com>

This file is part of MyRobotLab.

version 3 of the License, or (at your option) any later version. This program is...
Optimization Procedure

Initial Prompt $p_{init}$

Generate a code snippet that serves as a header comment for...

$LCS(p_{init}, d_{suffix}): 0.08 (\downarrow)$

Generated Completion

that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer.

$LCS(M(p_{init}), d_{suffix}): 0.21 (\uparrow)$

Attacker LLM $M'$

Rejection Sampling with Objective Function

$$\theta = \alpha \cdot LCS(M(p), d_{suffix}) + (1 - \alpha) \cdot -LCS(p, d_{suffix})$$

Optimized Prompt $p^*$

I have a question in file, How should I properly structure the header...

$LCS(p^*, d_{suffix}): 0.08 (\downarrow)$

Generated Completion

# MyRobotLab - Copyright (C) 2023 Author Name
<author@example.com>

This file is part of MyRobotLab.

version 3 of the License, or
(at your option) any later version. This program is

$LCS(M(p^*), d_{suffix}): 0.74 (\uparrow)$
Does this really work?
Evaluation Setup

Baselines

• **Prefix-Suffix** method (Carlini et al. 2022, Nasr et al. 2023, Bidderman et al. 2023): Uses pre-training data prefix directly, Blackbox

• **GCG** (Zou et al., 2023): Prompt optimization starting from pre-training data prefixes, white box

• **Reverse LM** (Pfau et al., 2023): Prompt optimization using Pythia 160m, blackbox
Evaluation Setup
Models, data and metrics

• Models:
  • Target (victim) Models: Alpaca, vicuna, Tulu, Olmo, Falcon
  • Attacker Models: Zephyr (Mistral-based model) and GPT4

• Pre-training data subsets (at lens 200, 300 and 500 tokens):
  • Redpajama: C4, CC, Arxiv, Books, Github (15k samples)
  • Dolma (16k samples)
  • RefinedWeb (3k samples)

• Metrics: Rouge-L between generation and target sequence
How do we fare against the baselines?
Let’s start with P-S on Tulu 7B, sequence length of 500 tokens, Rouge-l

We significantly outperform other baselines.
How do we fare against the baselines?

Let’s start with P-S on Tulu 7B, sequence length of 500 tokens, Rouge-l

<table>
<thead>
<tr>
<th></th>
<th>Github</th>
<th>ArXiv</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mem</td>
<td>LCS&lt;sub&gt;P&lt;/sub&gt;</td>
<td>Dis</td>
<td>Mem</td>
</tr>
<tr>
<td>↑</td>
<td>↓</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>P-S-Inst</td>
<td>0.247</td>
<td>0.124</td>
<td>-</td>
</tr>
<tr>
<td>Reverse-LM</td>
<td>0.233</td>
<td>0.204</td>
<td>0.833</td>
</tr>
<tr>
<td>Ours</td>
<td>0.363</td>
<td>0.129</td>
<td>0.814</td>
</tr>
</tbody>
</table>

We significantly outperform other baselines.

Github has the highest increase in memorization score.
How do we fare against the baselines?

Let’s start with P-S on Tulu 7B, sequence length of 500 tokens, Rouge-l

<table>
<thead>
<tr>
<th></th>
<th>Github</th>
<th>ArXiv</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mem</td>
<td>LCS&lt;sub&gt;P&lt;/sub&gt;</td>
<td>Dis</td>
</tr>
<tr>
<td>P-S-Inst</td>
<td>.247</td>
<td>.124</td>
<td>-</td>
</tr>
<tr>
<td>Reverse-LM</td>
<td>.233</td>
<td>.204</td>
<td>.833</td>
</tr>
<tr>
<td>Ours</td>
<td>.363</td>
<td>.129</td>
<td>.814</td>
</tr>
</tbody>
</table>

We significantly outperform other baselines.

Github has the highest increase in memorization score

Memorization scores on average: Tulu >> Vicuna > Alpaca
How do we fare against the baselines?

Now, let’s look at the base model, Llama

<table>
<thead>
<tr>
<th></th>
<th>Github</th>
<th>ArXiv</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mem</td>
<td>LCS$_P$</td>
<td>Dis</td>
</tr>
<tr>
<td></td>
<td>↑</td>
<td>↓</td>
<td>↑</td>
</tr>
<tr>
<td>P-S-Inst</td>
<td>.247</td>
<td>.124</td>
<td>-</td>
</tr>
<tr>
<td>Reverse-LM</td>
<td>.233</td>
<td>.204</td>
<td>.833</td>
</tr>
<tr>
<td>Ours</td>
<td>.363</td>
<td>.129</td>
<td>.814</td>
</tr>
<tr>
<td>P-S-Base</td>
<td>.263</td>
<td>.124</td>
<td>-</td>
</tr>
<tr>
<td>GCG</td>
<td>.265</td>
<td>.113</td>
<td>.435</td>
</tr>
</tbody>
</table>

We outperform baselines that assume access to the base model (Llama)
What if we don’t have access to the entire sequence?

Apart from GitHub, for the other domains limiting access does not diminish performance much.
Analysis: what is the best attacker?

Zephyr-7B can be even better than GPT-4 as an attacker!
Analysis: What are we extracting?

- We successfully extract 10.3% of the PII in the pre-training data subsets that we study, 1.4X more than the 4.2% of P-S.
- MIA-esque comparison: We see 30% more improvement over members, compared to non-members.
ACT V: Conclusion and what’s next?

“So, short story long.”
Conclusion

• We introduce a prompt optimization method to analyze how instruction-tuned LLMs memorize pre-training data, using instruction-based prompts.
Conclusion

• We introduce a **prompt optimization** method to analyze how instruction-tuned LLMs memorize pre-training data, using **instruction-based prompts**.

• Our findings indicate that **instruction-tuned models can show higher memorization** levels than what we expected!
Conclusion

- We introduce a prompt optimization method to analyze how instruction-tuned LLMs memorize pre-training data, using instruction-based prompts.
- Our findings indicate that instruction-tuned models can show higher memorization levels than what we expected!
- This increase does not necessarily imply that these models memorize/regurgitate more data or are more vulnerable, it just demonstrates a new attack vector!
Future Directions

• We need different memorization metrics, and we are on a good trajectory!
  • Compression metric
  • Recitation, recollection, reconstruction
  • Reasoning vs. reciting

Can we predict memorization?
Future Directions

• We need **different memorization metrics**, and we are on a good trajectory!
  • Compression metric
  • Recitation, recollection, reconstruction
  • Reasoning vs. reciting
• We need **more adversarial methods**, automated red-teaming!
• We need to consider **task complexity** as well!
• Can we **predict memorization**?