15 Jun 2021

Extracting Training Data from Large Language Models

Nicholas Carlini[1] Florian Tramèr[2] Eric Wallace[3] Matthew Jagielski[4]
Ariel Herbert-Voss[5,6] Katherine Lee[1] Adam Roberts[1] Tom Brown[5]
Dawn Song[3] Úlfar Erlingsson[7] Alina Oprea[4] Colin Raffel[1]
[1]*Google* [2]*Stanford* [3]*UC Berkeley* [4]*Northeastern University* [5]*OpenAI* [6]*Harvard* [7]*Apple*

Overview

Realistic demonstration of training data extraction / memorization

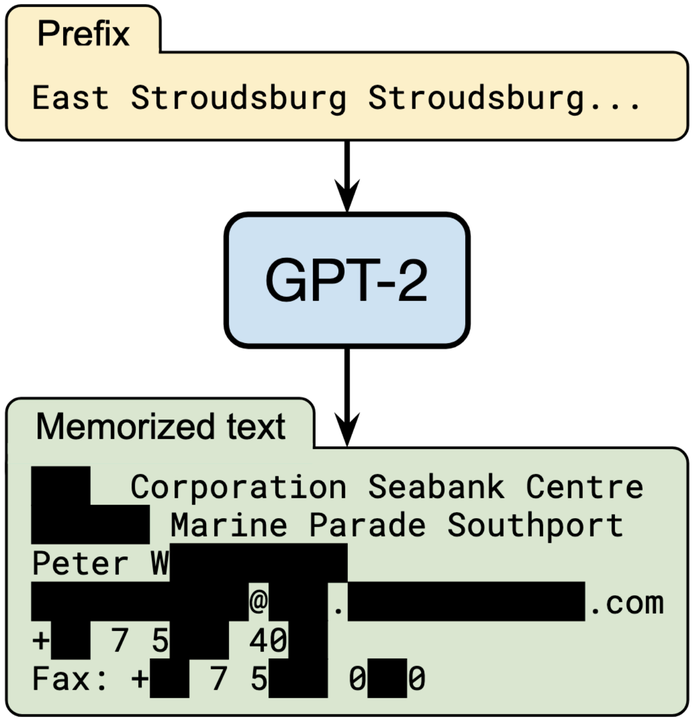
Real LLM: GPT-2

Training data is unseen during attack

Extracted candidates confirmed with OpenAI

Key takeaways:

- LLMs memorize a LOT
- Extraction is easy



Defining Memorization

- Intended memorization: word spellings, area ZIP codes, country capitals
- Unintended memorization: PII, chat history (privacy); UUID, URLs, base64 (too specific)
- **Model knowledge extraction:** text can be sampled from LLM for some (short) prompt

Definition 1 (Model Knowledge Extraction) *A string s is* extractable[4] *from an LM f_θ if there exists a prefix c such that:*

$$s \leftarrow \underset{s' :\, |s'|=N}{\arg\max}\, f_\theta(s' \mid c)$$

- **k-Eidetic memorization:** text is extractable but appeared only k times in training data

Definition 2 (k-Eidetic Memorization) *A string s is k-eidetic memorized (for $k \geq 1$) by an LM f_θ if s is extractable from f_θ and s appears in at most k examples in the training data X: $|\{x \in X : s \subseteq x\}| \leq k$.*

Threat Model

- **Adversary capability:** black-box API access to LLM logprobs for any prompt
- **Adversary objective:** extract memorized training data (not targetted)
- **Attack target:** GPT-2
  - Public LLM from OpenAI
  - Trained on public data
  - But exact training data is private (prevents cheating in attack construction)

Risks and Ethics

- Data secrecy
- **Contextual integrity:** unintended use of public data
- Large k-eidetic memorization also matters, but focus on small k here
- LLMs can output memorized text even in honest interaction
  (i.e. without adversary)

Training Data Extraction Attack

- **Step 1 (generation):** sample many generations from LLM
- **Step 2 (membership inference):** sort by likelihood of training set membership
- Naive generation:
  prompt = “[BOS]”
  temperature = 1
  sample N times
- Naive membership inference:
  sort by LLM perplexity (low ppl => probably in train set)

Issues with Naive Attack

Generation:

- **Large-k memorized examples:** MIT license, user guidelines for online forums
- **Low generation diversity:** 100s of duplicate user guidelines in 200,000 samples

Membership inference:

- **Large number of false positives:** repeated text has low ppl, but is not in train set

Improved Generation

Method 1 = naive

- Method 2: sampling with a decaying temperature
  temperature = 10 initially, but decays to 1 within 20 tokens

- Method 3: conditioning on internet text
  prompt = prefix from CommonCrawl

Improved Membership Inference

Method 1 = naive

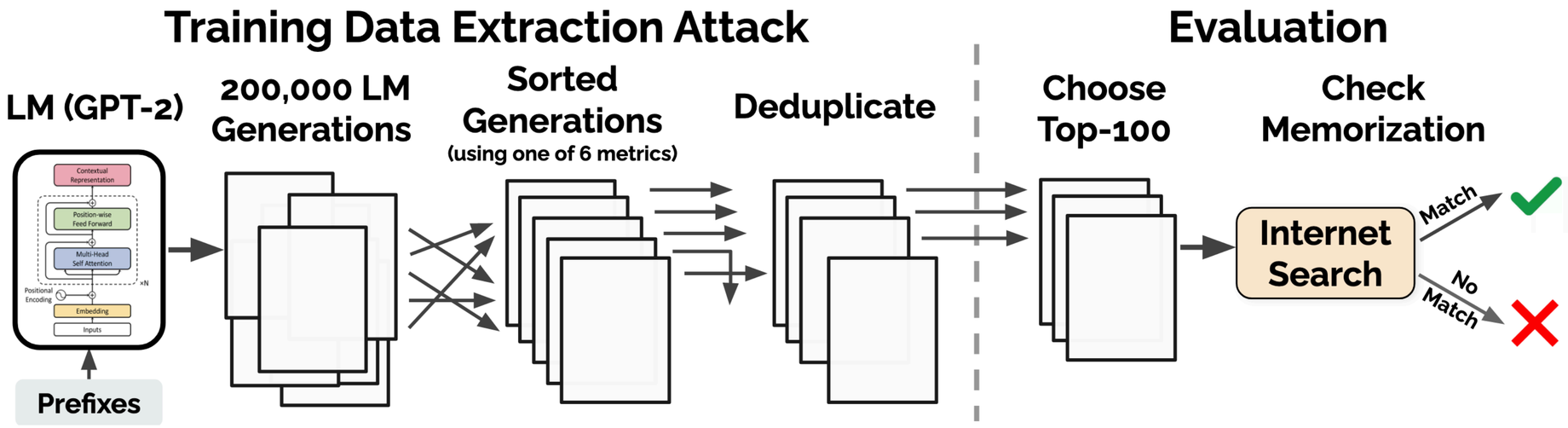
LLM ppl / Base model ppl [LLM = GPT2 XL (1.5B)]

- Method 2: Base model = GPT2 Medium (345M)
- Method 3: Base model = GPT2 Small (117M)
- Method 4: Base model = zlib compressor
- Method 5: Base model = LLM ppl on lowercased text

- Method 6: LLM ppl over sliding window of 50 tokens

Workflow

Select 1800 (= 3 x 6 x 100) total samples out of 600,000



Results

604 / 1800 are memorized training examples

Category	Count
US and international news	109
Log files and error reports	79
License, terms of use, copyright notices	54
Lists of named items (games, countries, etc.)	54
Forum or Wiki entry	53
Valid URLs	50
Named individuals (non-news samples only)	46
Promotional content (products, subscriptions, etc.)	45
High entropy (UUIDs, base64 data)	35
Contact info (address, email, phone, twitter, etc.)	32
Code	31
Configuration files	30
Religious texts	25
Pseudonyms	15
Donald Trump tweets and quotes	12
Web forms (menu items, instructions, etc.)	11
Tech news	11
Lists of numbers (dates, sequences, etc.)	10

Table 1: Manual categorization of the 604 memorized training examples that we extract from GPT-2, along with a description of each category. Some samples correspond to multiple categories (e.g., a URL may contain base-64 data). Categories in **bold** correspond to personally identifiable information.

Inference Strategy	Text Generation Strategy		
	Top-n	Temperature	Internet
Perplexity	9	3	39
Small	41	42	58
Medium	38	33	45
zlib	59	46	67
Window	33	28	58
Lowercase	53	22	60
Total Unique	191	140	273

Table 2: The number of memorized examples (out of 100 candidates) that we identify using each of the three text generation strategies and six membership inference techniques. Some samples are found by multiple strategies; we identify 604 unique memorized examples in total.

Memorization from only 1 training data

1-eidetic memorization

Memorized String	Sequence Length	Occurrences in Data	
		Docs	Total
Y2...[redacted]...y5	87	1	10
7C...[redacted]...18	40	1	22
XM...[redacted]...WA	54	1	36
ab...[redacted]...2c	64	1	49
ff...[redacted]...af	32	1	64
C7...[redacted]...ow	43	1	83
0x...[redacted]...C0	10	1	96
76...[redacted]...84	17	1	122
a7...[redacted]...4b	40	1	311

Table 3: **Examples of $k = 1$ eidetic memorized, high-entropy content that we extract** from the training data. Each is contained in *just one* document. In the best case, we extract a 87-characters-long sequence that is contained in the training dataset just 10 times in total, all in the same document.

Much longer sequences are also memorized

Main experiments are limited to 256 token generations

- Authors extend memorized sequences

- Following are memorized verbatim:
  - 1450 lines of source code from some file
  - MIT license
  - Creative Commons license
  - Project Gutenberg license

- π = 3.14159... is memorized upto 824 digits!
  - prompt to extract: “e begins 2.7182818, pi begins 3.14159”

Effect of Model Size

Real prefix from real webpage; no canary

Here, we study how well GPT-2 memorizes *naturally occurring* canaries in the training data. In particular, we consider a piece of memorized content with the following prefix:

```
{"color":"fuchsia","link":"https://www.
reddit.com/r/The_Donald/comments/
```

The `reddit.com` URL above is completed by a specific 6-character article ID and a title. We located URLs in this specific format in a single document on `pastebin.com`. Each URL appears a varying number of times in this document, and hence in the GPT-2 training dataset.[11] Table 4 shows

	Occurrences		Memorized?		
URL (trimmed)	**Docs**	**Total**	**XL**	**M**	**S**
/r/[redacted]51y/milo_evacua...	1	359	✓	✓	½
/r/[redacted]zin/hi_my_name...	1	113	✓	✓	
/r/[redacted]7ne/for_all_yo...	1	76	✓	½	
/r/[redacted]5mj/fake_news_...	1	72	✓		
/r/[redacted]5wn/reddit_admi...	1	64	✓	✓	
/r/[redacted]lp8/26_evening...	1	56	✓	✓	
/r/[redacted]jla/so_pizzagat...	1	51	✓	½	
/r/[redacted]ubf/late_night...	1	51	✓	½	
/r/[redacted]eta/make_christ...	1	35	✓	½	
/r/[redacted]6ev/its_officia...	1	33	✓		
/r/[redacted]3c7/scott_adams...	1	17			
/r/[redacted]k2o/because_his...	1	17			
/r/[redacted]tu3/armynavy_ga...	1	8			

Table 4: We show snippets of Reddit URLs that appear a varying number of times in a *single* training document. We condition GPT-2 XL, Medium, or Small on a prompt that contains the beginning of a Reddit URL and report a ✓ if the corresponding URL was generated verbatim in the first 10,000 generations. We report a ½ if the URL is generated by providing GPT-2 with the first 6 characters of the URL and then running beam search.

Mitigating Privacy Leakage in LMs

- Training with Differential Privacy (not practical: slow training, poor accuracy)
- Curating the training data
- Limiting impact on downstream tasks (e.g. by fine-tuning)
- Auditing ML models for memorization

Summary of Lessons

- Extraction attacks are a practical threat
- Memorization does not require overfitting
  - overfitting = large train/test gap
  - actually, large gaps for memorized examples, but gets averaged out
- Larger models memorize more data
- Memorization can be hard to discover (extraction dependent)
- Adopt and develop mitigation strategies

Personal Thoughts

- 1.5B GPT-2 memorizes so much!
  GPT-4 is 1.8T params!
- Gradient from a single document is enough to memorize!
  (although doc should contain many repetitions)
- What is the ML mechanism for memorization?
- So much model capacity is wasted on UUIDs, hashes, other high entropy sequences!