Extracting Training Data from Large Language Models

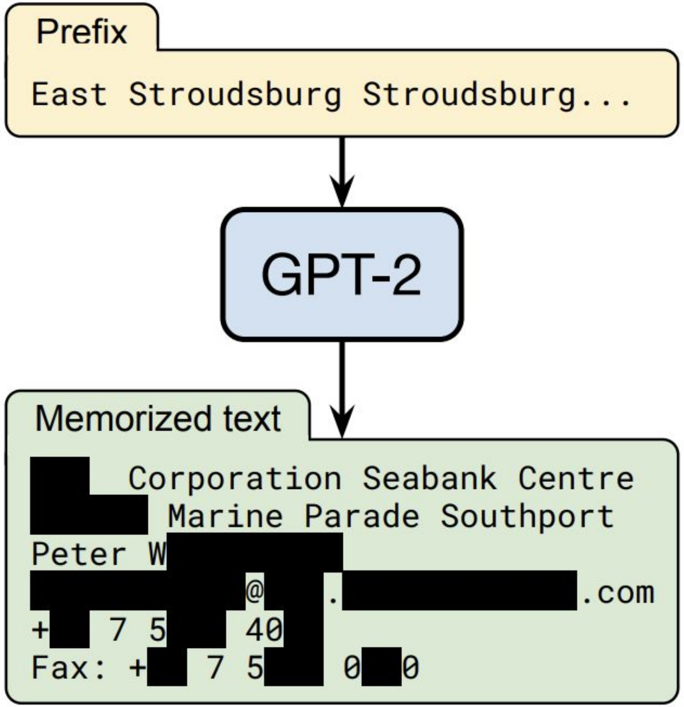
Abstract

- Training data extraction attack on large language models
- Recovering individual training examples by querying the model
- Attack on GPT-2, successfully extracting verbatim text sequences
- Larger models are more vulnerable to the attack
- Discussing lessons learned and safeguards for training large language models

Introduction

- Role of language models in natural language processing tasks
- Increased size and training data of modern neural network-based models
- Privacy concerns with language models exposing training data information
- Membership inference attacks and belief about memorization
- Challenging the belief by demonstrating memorization in large language models

Introduction



Prefix

East Stroudsburg Stroudsburg...

GPT-2

Memorized text

[redacted] Corporation Seabank Centre
[redacted] Marine Parade Southport
Peter W[redacted]
[redacted]@[redacted].[redacted].com
+[redacted] 7 5[redacted] 40[redacted]
Fax: +[redacted] 7 5[redacted] 0[redacted]0

Figure 1: **Our extraction attack.** Given query access to a neural network language model, we extract an individual person's name, email address, phone number, fax number, and physical address. The example in this figure shows information that is all accurate so we redact it to protect privacy.

Background & Related Work

- Overview of large neural network-based language models
- Language modeling as a fundamental component in NLP pipelines
- Training objectives and minimizing the loss function
- Focus on GPT-2, a variant of Transformer LMs trained on web data
- Concerns about training data privacy and related attacks
- Comparison with state-of-the-art LMs and privacy-preserving techniques

Threat Model & Ethics

- Practicality of training data extraction attacks
- Definition of "memorization" in language models
- Threat model with black-box access to the model
- Risks of training data extraction and privacy concerns
- Ethical considerations and responsible disclosure

Initial Training Data Extraction Attack

- Baseline approach for extracting training data
- Two-step procedure: text generation and membership inference
- Initial text generation scheme with top-n sampling strategy
- Initial membership inference using perplexity as a measure
- Results and limitations of the baseline attack

Improved Training Data Extraction Attack

- Improved approach to address limitations of the previous attack
- Improved text generation schemes: decaying temperature and conditioning on Internet text
- Enhanced membership inference methods: comparison-based metrics and entropy quantification
- Sliding window approach for handling uncertainty within context
- Objectives and effectiveness of the improvements

Improved Training Data Extraction Attack

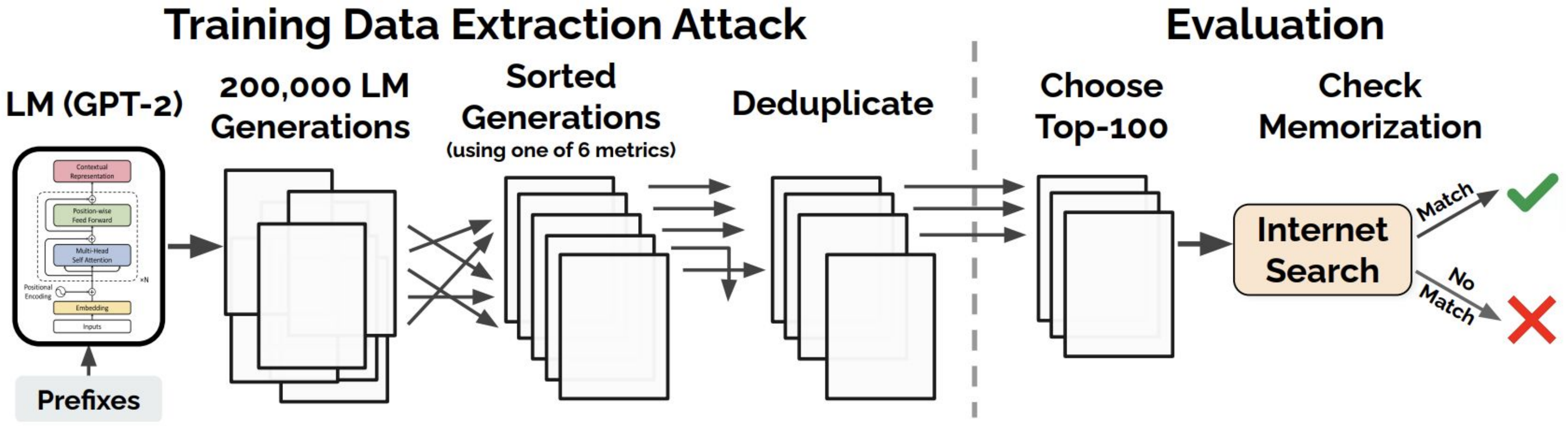
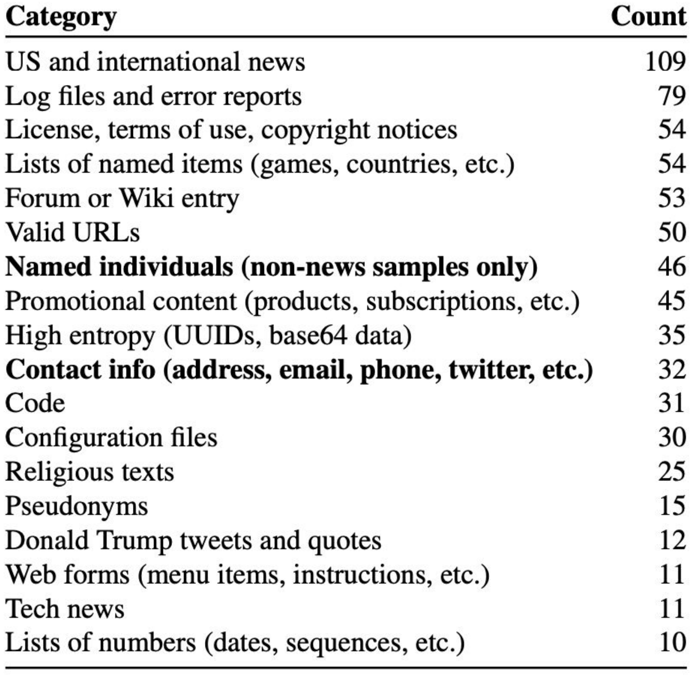


Figure 2: **Workflow of our extraction attack and evaluation. 1) Attack.** We begin by generating many samples from GPT-2 when the model is conditioned on (potentially empty) prefixes. We then sort each generation according to one of six metrics and remove the duplicates. This gives us a set of potentially memorized training examples. **2) Evaluation.** We manually inspect 100 of the top-1000 generations for each metric. We mark each generation as either memorized or not-memorized by manually searching online, and we confirm these findings by working with OpenAI to query the original training data.

Evaluating Memorization

- Methodology for evaluating data extraction methods
- Construction of datasets and ordering based on membership inference metrics
- Manual inspection and validation of selected samples
- Results of the evaluation and categorization of memorized content
- Examples of different categories and extraction of longer verbatim sequences

Evaluating Memorization



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.

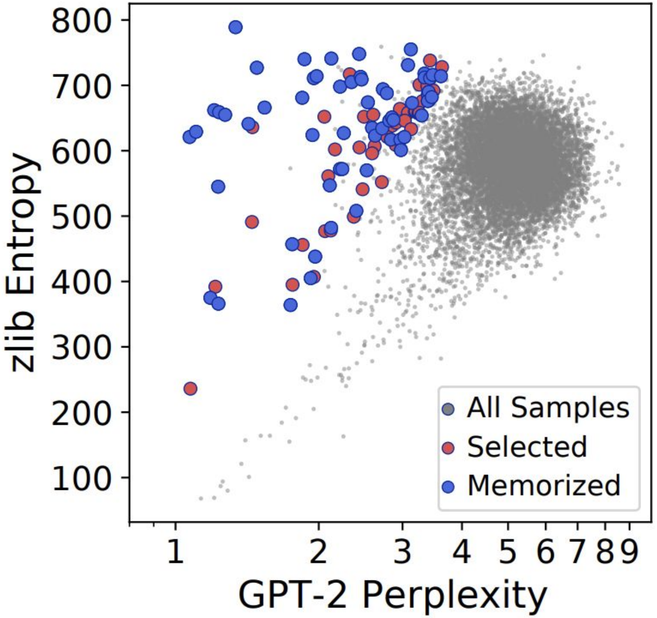
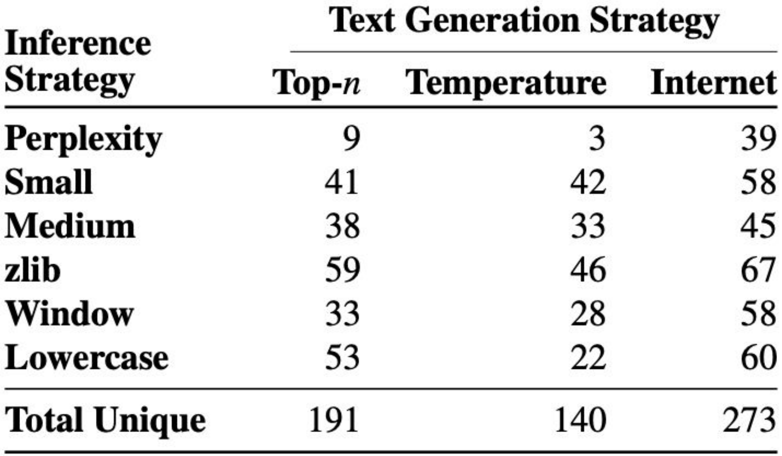


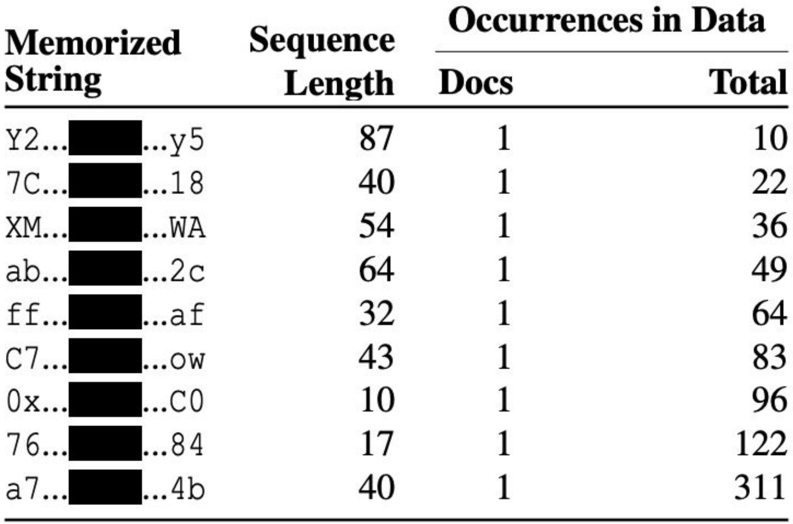
Figure 3: The zlib entropy and the perplexity of GPT-2 XL for 200,000 samples generated with top-*n* sampling. In red, we show the 100 samples that were selected for manual inspection. In blue, we show the 59 samples that were confirmed as memorized text. Additional plots for other text generation and detection strategies are in Figure 4.

Evaluating Memorization



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.



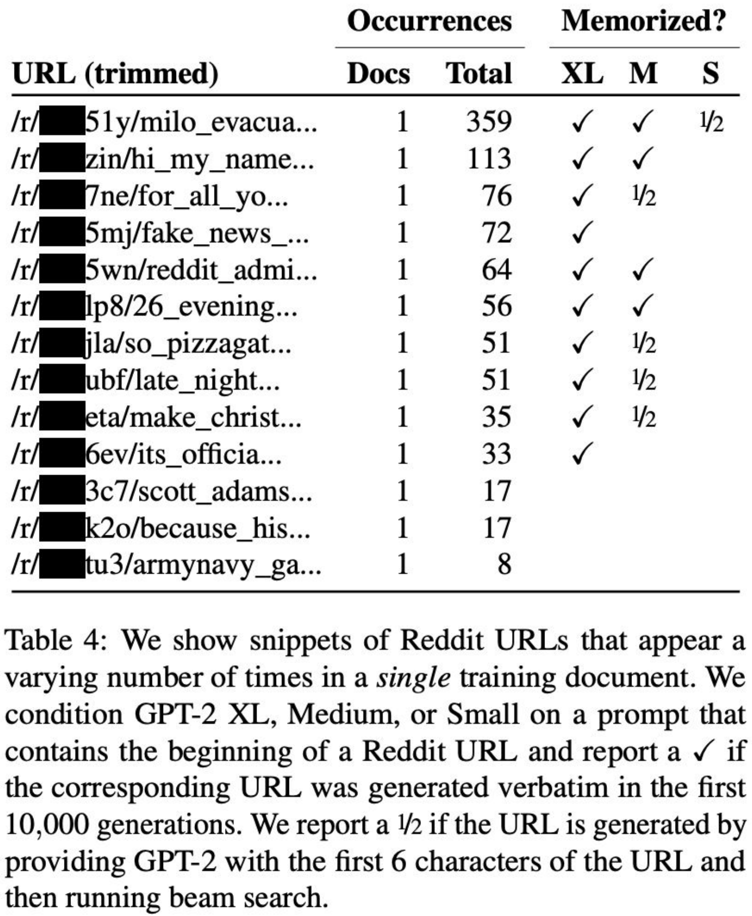
Memorized String	Sequence Length	Occurrences in Data	
		Docs	Total
`Y2...`[redacted]`...y5`	87	1	10
`7C...`[redacted]`...18`	40	1	22
`XM...`[redacted]`...WA`	54	1	36
`ab...`[redacted]`...2c`	64	1	49
`ff...`[redacted]`...af`	32	1	64
`C7...`[redacted]`...ow`	43	1	83
`0x...`[redacted]`...C0`	10	1	96
`76...`[redacted]`...84`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

Correlating Memorization with Model Size & Insertion Frequency

- Correlation between memorization, model size, and insertion frequency
- Examination of naturally occurring canaries and GPT-2's memorization
- Analysis using Reddit URLs as canaries
- Results showing larger models memorize more data and risk of memorizing sensitive information

Correlating Memorization with Model Size & Insertion Frequency



	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

- Strategies to mitigate privacy risks associated with memorized training data
- Training models with differential privacy and challenges with web data
- Curating training data and de-duplication as additional strategies
- Limiting impact on downstream applications and auditing for memorization
- Proposed mitigation strategies and complementing theoretical privacy bounds with empirical audits

Lessons and Future Work

- Extraction attacks as a practical threat
- Memorization not requiring overfitting and importance of understanding underlying reasons
- Correlation between memorization and model size
- Difficulty in discovering memorized content and exploring better prefix selection
- Adoption and development of mitigation strategies

Conclusion

- Addressing training data memorization for large language model adoption
- Demonstrating efficient extraction attacks on GPT-2
- Applicability of attacks to any language model
- Increasing vulnerabilities with larger language models
- Need for specific techniques to tackle memorization attacks
- Importance of differential privacy training methods at extreme scales
- Further research to understand causes and dangers of memorization
- Exploration of preventive measures

Key Takeaways

- Language models can be susceptible to training data extraction attacks
- Large models can memorize and leak individual training examples
- Privacy concerns arise from the disclosure of sensitive information
- Mitigation strategies include differential privacy training and data curation
- Further research is needed to develop effective preventive measures

Thank You!

- Questions?