Rethinking LLM Memorization through the Lens of Adversarial Compression

Anonymous Authors[1]

1. Introduction

A central question in the discussion of large language models (LLMs) concerns the extent to which they *memorize* their training data versus how they *generalize* to new tasks and settings. Most practitioners seem to (at least informally) believe that LLMs do some degree of both: they *clearly* memorize parts of the training data—for example, are often able to reproduce large portions of training data verbatim (Carlini et al., 2023)—but they also seem to learn from this data, allowing them to generalize to new settings. The precise extent to which they do one or the other has massive implications for the practical and legal aspects of such models (Cooper et al., 2023). Do LLMs truly produce new content, or do they only remix their training data? Should the act of training on copyrighted data be deemed unfair use of data, or should fair use be judged by the model's memorization? With respect to people, we distinguish plagiarising content from learning from it, but how should this extend to LLMs? The answer to such questions inherently relates to the extent to which LLMs memorize their training data.

However, even defining memorization for LLMs is challenging and many existing definitions leave a lot to be desired. In this work, we propose a new definition of memorization based on a compression argument. Our definition posits that *a phrase present in the training data is memorized if we can make the model reproduce the phrase using a prompt (much) shorter than the phrase itself*. Operationalizing this definition requires finding the shortest adversarial input prompt that is specifically optimized to produce a target output. We call this ratio of input to output tokens the Adversarial Compression Ratio (ACR). In other words, memorization is inherently tied to whether a certain output can be represented in a *compressed* form, beyond what language models can do with typical text. We argue that such a definition provides an intuitive notion of memorization—if a certain phrase exists within the LLM training data (e.g., is not itself generated text) *and* it can be reproduced with fewer input tokens than output tokens, then the phrase must be stored somehow within the weights of the LLM. Although it may be more natural to consider compression in terms of the LLM-based notions of input/output perplexity, we argue that a simple compression ratio based on input/output token counts provides a more intuitive explanation to non-technical audiences, and has the potential to serve as a legal basis for important questions about memorization and permissible data use. In addition to its intuitive nature, our definition has several other desirable qualities. We show that it appropriately ascribes many famous quotes as being memorized by existing LLMs (i.e. they have high ACR values). On the other hand, we find that text not in the training data of an LLM, such as samples posted on the internet after the training period, are not compressible, that is their ACR is low.

We examine several unlearning methods using ACR to show that they do not substantially affect the memorization of the model. That is, even after explicit finetuning, models asked to "forget" certain pieces of content are still able to reproduce them with a high ACR—in fact, not much smaller than with the original model. Our approach provides a simple and practical perspective on what memorization can mean, providing a useful tool for functional and legal analysis of LLMs.

2. Why We Need A New Definition

With LLMs ingesting more and more data, questions about their memorization are attracting attention (e.g. Carlini et al., 2019; 2023; Nasr et al., 2023; Zhang et al., 2023). There remains a pressing need to accurately define memorization in a way that serves as a practical tool to ascertain the fair use of public data from a legal standpoint. To ground the problem, consider the court's role in determining whether an LLM is breaching copyright. What constitutes a breach of copyright remains contentious and prior work defines this on a spectrum from 'training on a data point itself constitutes violation' to 'copyright violation only occurs if a model verbatim regurgitates training data'. To formalize our argument for a new notion of memorization, we start with three definitions from prior work to highlight some of

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the gaps in the current thinking about memorization.

Discoverable memorization (Carlini et al., 2023), which says a string is memorized if the first few words elicit the rest of the quote exactly, has three particular problems. It is very permissive, easy to evade, and requires validation data to set parameters. Another notion is Extractable Memorization (Nasr et al., 2023), which says that if *there exists* a prompt that elicits the string in response. This falls too far on the other side of the issue by being **very restrictive**—what if the prompt includes the entire string in question, or worse, the instructions to repeat it? LLMs that are good at repeating will follow that instruction and output any string they are asked to. The risk is that it is possible to label any element of the training set as memorized, rendering this definition unfit in practice. Another definition is Counterfactual Memorization (Zhang et al., 2023), which aims to separate memorization from generalization and is tested through retraining many LLMs. Given the cost of training LLMs, such a definition is **impractical** for legal use.

In addition to these definitions from prior work on LLM memorization, there are several other seemingly viable approaches to memorization. Ultimately, we argue all of these frameworks—the definitions in existing work and the approaches described below—are each missing key elements of a good definition for assessing fair use of data.

Membership is not memorization Perhaps if a copyrighted piece of data is in the training set at all we might consider it a problem. However, there is a subtle but crucial difference between training set membership and memorization. In particular, the ongoing lawsuits in the field (e.g. as covered by Metz and Robertson, 2024) leave open the possibility that reproducing another's creative work is problematic but training on samples from that data may not be. This is common practice in the arts—consider that a copycat comedian telling someone else's jokes is stealing, but an up-and-comer learning from tapes of the greats is doing nothing wrong. So while membership inference attacks (MIAs) (e.g. Shokri et al., 2017) may look like tests for memorization and they are even intimately related to auditing machine unlearning (Carlini et al., 2021; Pawelczyk et al., 2023; Choi et al., 2024), they have three issues as tests for memorization: It is very restrictive, it is hard to arbitrate and evaluation techiniques are brittle.

3. Adversarial Compression Ratio

Our definition of memorization is based on answering the following question: Given a piece of text, how short is the minimal prompt that elicits that text exactly? In this section, we formally define and introduce our MINIPROMPT algorithm that we use to answer our central question.

To begin, let a target natural text string s have a token sequence representation $x \in \mathcal{V}^*$ which is a list of integer-valued indices that index a given vocabulary $\mathcal{V}$. We use $|\cdot|$ to count the length of a token sequence. A tokenizer $T : s \mapsto x$ maps from strings to token sequences. Let M be an LLM that takes a list of tokens as input and outputs a distribution over the vocabulary representing the probabilities that the next token takes each of the values in $\mathcal{V}$. Consider that M can perform generation by repeatedly predicting the next token from all the previous tokens with the `argmax` of its output appended to the sequence at each step (this process is called greedy decoding). With a slight abuse of notation, we will also call the greedy decoding result the output of M. Let y be the token sequence generated by M, which we call a completion or response: $y = M(x)$, which in natural language says that the model generates y when prompted with x or that x elicits y as a response from M. So our compression ratio ACR is defined for a target sequence y as $\text{ACR}(M, y) = \frac{|y|}{|x^*|}$, where $x^* = \arg\min_x |x|$ s.t. $M(x) = y$.

Definition 1 (τ-Compressible Memorization)**.** *Given a generative model M, a sample y from the training data is τ-memorized if the* $\text{ACR}(M, y) > \tau(y)$.

The threshold $\tau(y)$ is a configurable parameter of this definition. We might choose to compare the ACR to the compression ratio of the text when run through a general-purpose compression program (explicitly assumed not to have memorized any such text) such as GZIP (Gailly and Adler, 1992) or SMAZ (Sanfilippo, 2006). This amounts to setting $\tau(y)$ equal to the SMAZ compression ratio of y, for example. Alternatively, one might even use the compression ratio of the arithmetic encoding under another LLM as a comparison point, for example if it was known with certainty that the LLM was never trained on the target output, and hence could not have memorized it (Delétang et al., 2023). In reality, copyright attribution cases are always subjective, and the goal of this work is not to argue for the right threshold function, rather to advocate for the adversarial compression framework for arbitrating fair data use. Thus, we use $\tau = 1$, which we believe has substantial practical value.[1]

Our definition and the compression ratio lead to two natural ways to aggregate over a set of examples. First, we can average the ratio over all samples/test strings and report the *average compression ratio* (this is τ-independent). Second, we can label samples with a ratio greater than one as *memorized* and discuss the *portion memorized* over some set of test cases (for our choice of $\tau = 1$).

[1]There exist prompts like "count from 1 to 1000," for which a chat model M is able to generate "$1, 2, \ldots, 1000$," which results in a very high ACR. However, for copyright purposes, we argue that this category of algorithmic prompts are in the gray area where determining memorization is difficult and beyond the scope of this paper given our primary application to creative works.

References

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. *arXiv preprint arXiv:2112.03570*, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023.

Dami Choi, Yonadav Shavit, and David K Duvenaud. Tools for verifying neural models' training data. *Advances in Neural Information Processing Systems*, 36, 2024.

A Feder Cooper, Katherine Lee, James Grimmelmann, Daphne Ippolito, Christopher Callison-Burch, Christopher A Choquette-Choo, Niloofar Mireshghallah, Miles Brundage, David Mimno, Madiha Zahrah Choksi, et al. Report of the 1st workshop on generative ai and law. *arXiv preprint arXiv:2311.06477*, 2023.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. Language modeling is compression. *arXiv preprint arXiv:2309.10668*, 2023.

Jean-Loup Gailly and Mark Adler. gzip. https://www.gnu.org/software/gzip/, 1992. Accessed: 2024-05-21.

Cade Metz and Katie Robertson. Openai seeks to dismiss parts of the new york times's lawsuit. *The New York Times*, 2024. URL https://www.nytimes.com/2024/02/27/technology/openai-new-york-times-lawsuit.html#:~:text=In%20its%20suit%2C%20The%20Times,someone%20to%20hack%20their%20chatbot.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

Salvatore Sanfilippo. Smaz: Small strings compression library. https://github.com/antirez/smaz, 2006. Accessed: 2024-05-21.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36: 39321–39362, 2023.